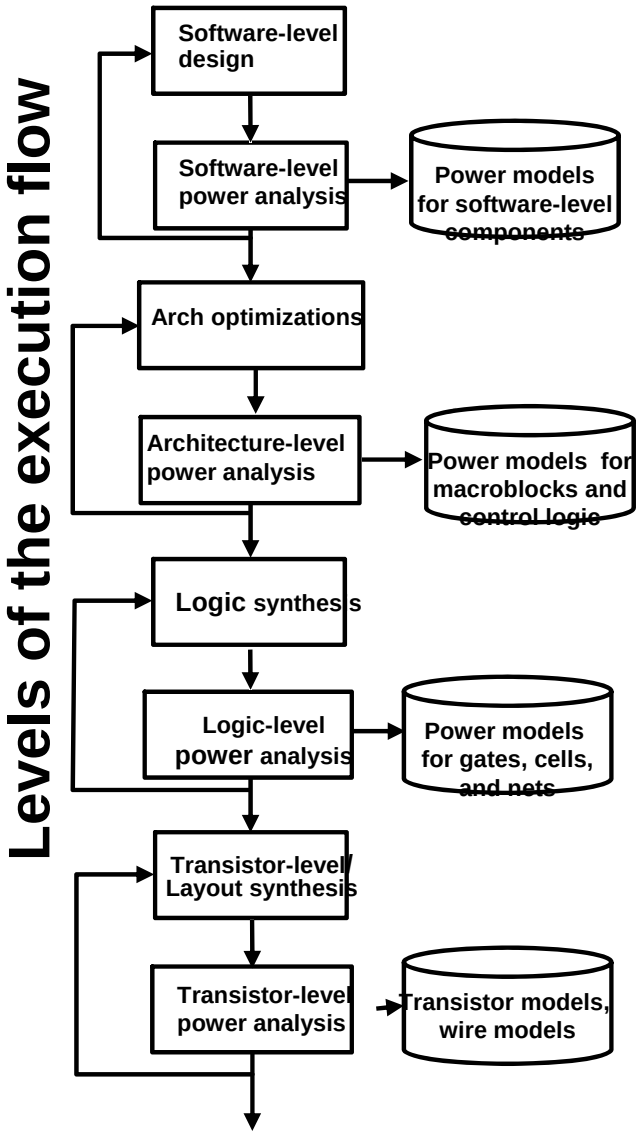
# Architecture level modelling

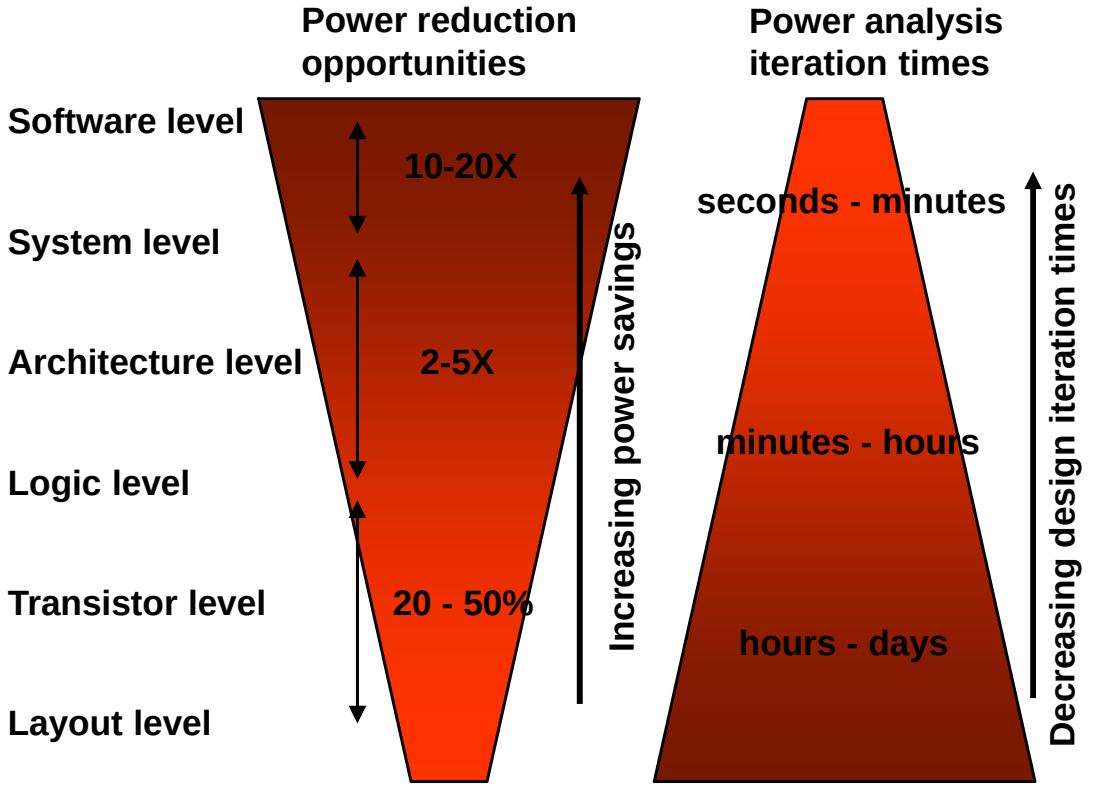
Ramon Canal

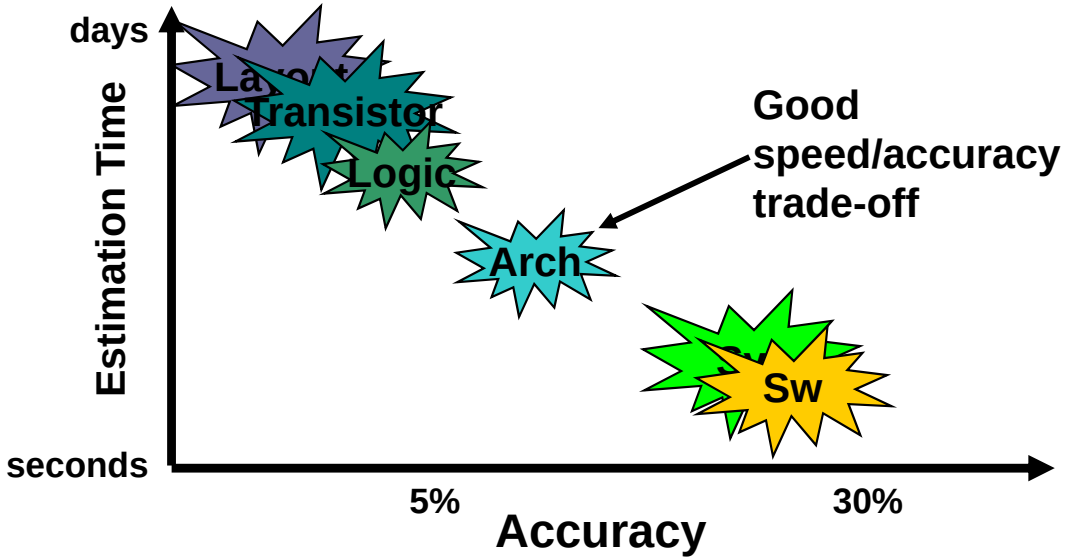
NCD - Master MIRI

Departament d'Arquitectura de Computadors
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# How Power Estimation is Addressed




Departament d'Arquitectura de Computadors
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# Agenda

- Introduction
- Using Hardware Counters
  - Based on PARAPET group work (Princeton)
- Architecture Simulation
- Statistical sampling
- Related Work
- Conclusions

# Advantages of Hdw Counters

- Power Models reflecting modern processors
  - Clock gating, power
  - Voltage regulation, di/dt
- Need for Fast-Realtime Modeling and Measurement to observe long time periods
  - Thermal time constants: O(s)
  - Not feasible even with architecural simulators
    - i.e.: 1s of real run ⇔~5 x IPC hrs of WATTCH simulation
- Need live, run-time power/thermal measures
  - Dynamic Thermal Management
  - Power-Aware OS & Systems control

# Where all this is useful?

- Measurement/Modeling for microarchitectural details
- Compiler level power
  - SW power profiling
- Power Aware OS
  - Dynamic power/thermal/March. Configuration
    - Dynamic memory allocate, Process cruise control, etc.
- Demonstrates modern processor power
- Need for speed! Long Timescales, thermal constants
- Identify program phases w/o knowledge of code, no basic block info whatsoever
- Program signatures for detailed simulation, say: “power points rather than simpoints”

# INTRODUCTION

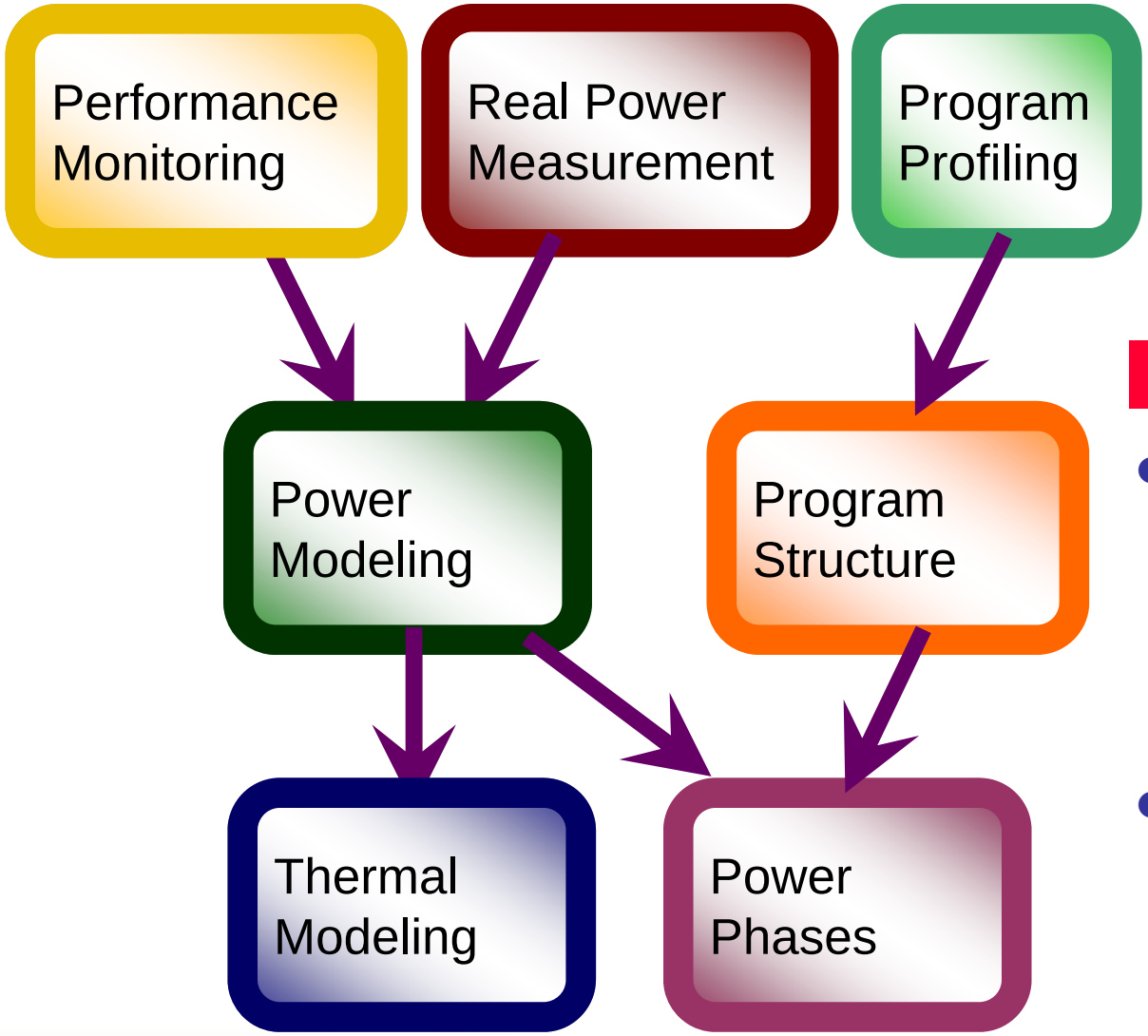
- Runtime processor power
  - Measurement with HW
  - Estimation with Performance counters
    - CPU Unit Power Breakdowns
    - Runtime verification

Detailed

- Processor thermal modeling
- Power Phase Behavior of programs
- Mapping between power behavior and program structure

Touch

# THE BIG PICTURE

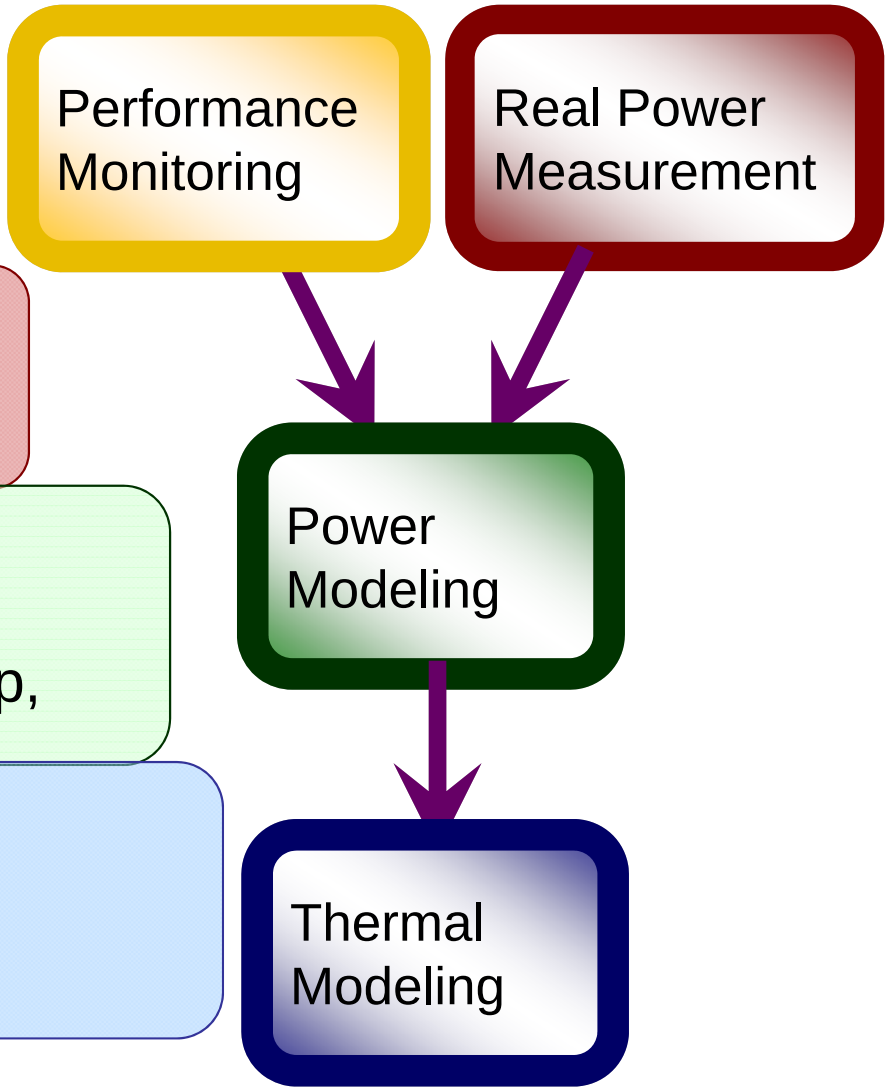



Bottom line…

- *To Estimate component power & temperature breakdowns for P4 at runtime…*
- *To analyze how power phase behavior relates to program structure*

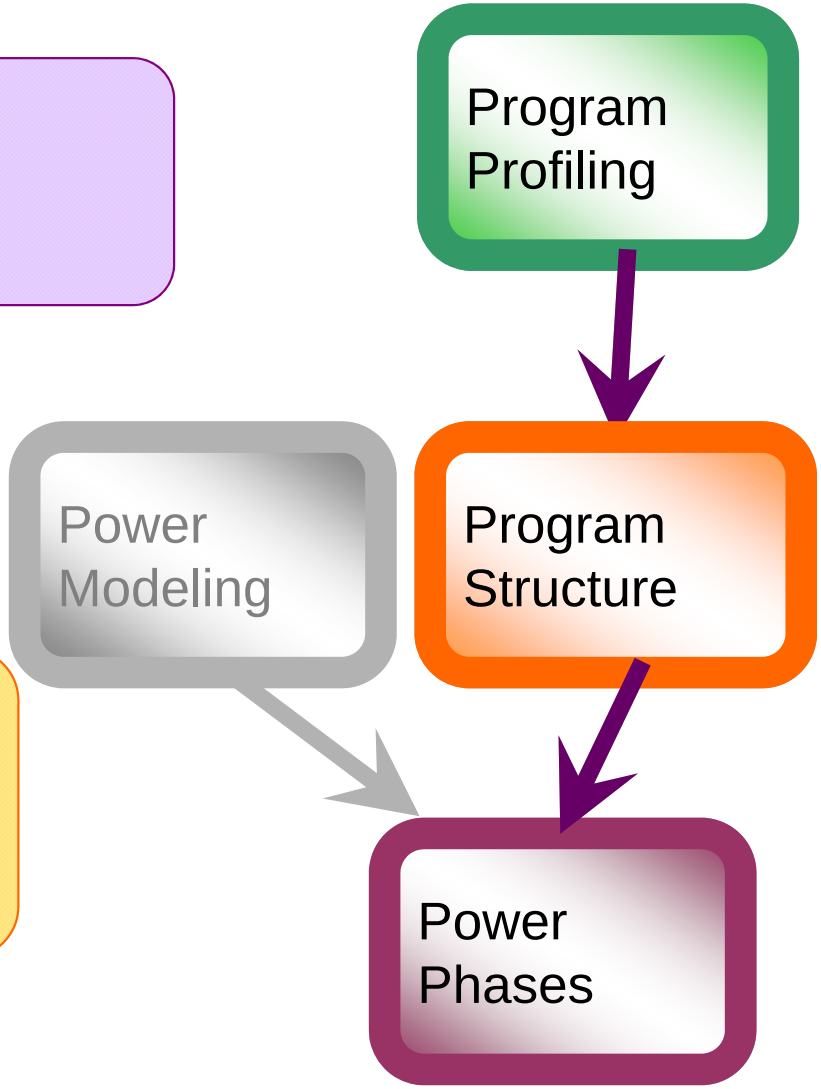
# Agenda

- Performance Monitoring
  - P4 Performance Counters
  - Performance Reader LKM
- Real Power Measurement
  - P4 Power Measurement Setup
  - Examples
- Power Modeling
  - P4 Power Model
  - Model + Measurement Sync Setup, Verification
- Thermal Modeling
  - Brief Thermal Model Intro
  - Ppro Thermal Model Results



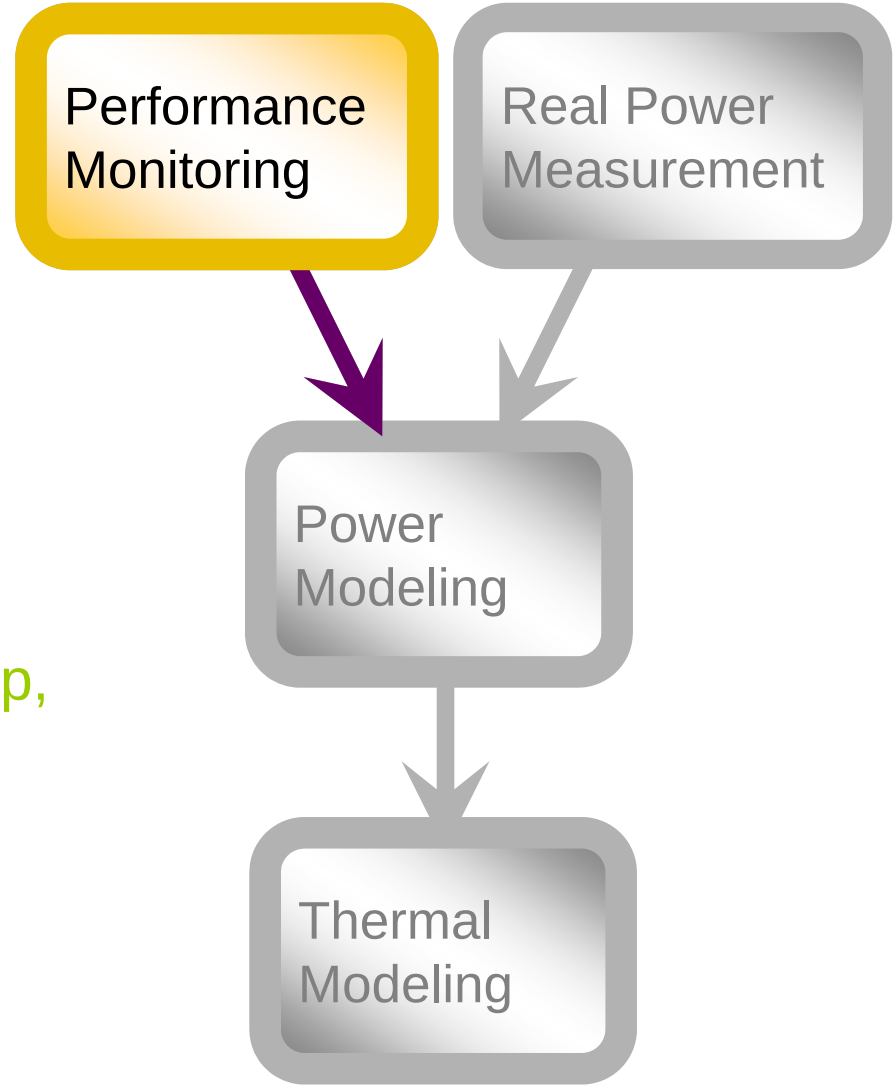
# Bonus Material

- Power Phase Behavior
  - Similarity Based on Power Vectors
  - Identifying similar program regions
- Profiling Execution Flow
  - Sampling process' execution
  - "PCsampler" LKM
- Program Structure
  - Execution vs. Code space
  - Power Phases ⇔ Exec. Phases
    - <OR VICE VERSA>

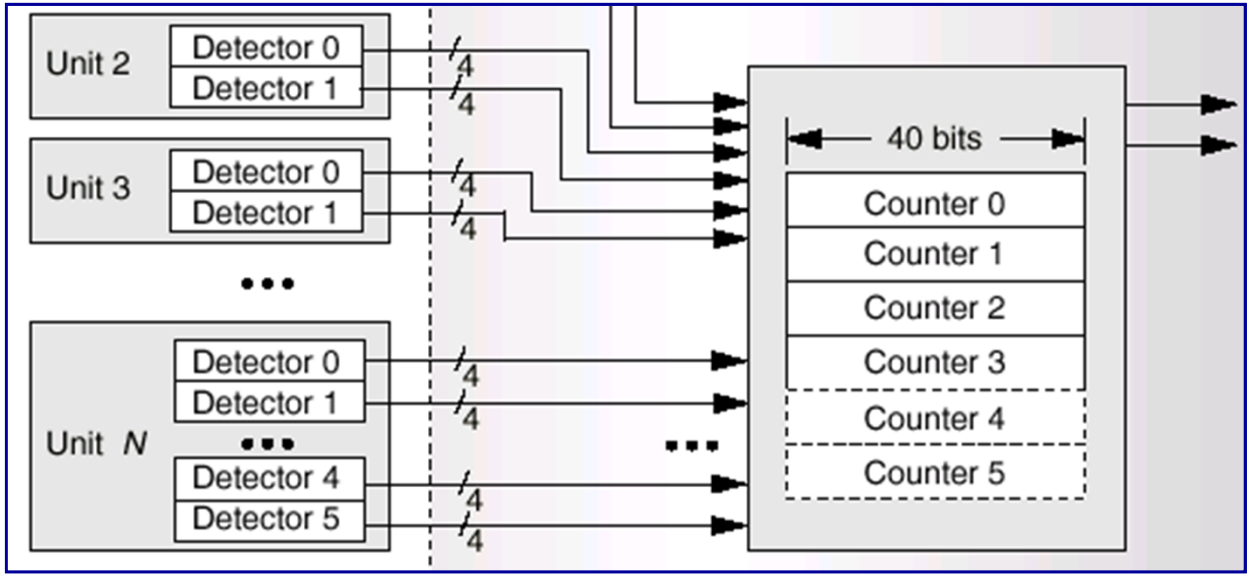


# Performance Monitoring

- Related Work
- Performance Monitoring
  - P4 Performance Counters
  - Performance Reader LKM
- Real Power Measurement
  - P4 Power Measurement Setup
  - Examples
- Power Modeling
  - P4 Power Model
  - Model + Measurement Sync Setup, Verification
- Thermal Modeling
  - Refined Thermal Model
  - Ex: Ppro Thermal Model

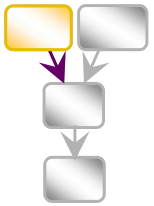


# Live CPU Performance Monitoring with Hardware Counters




- Most CPUs have hardware performance counters
- P4 Performance Monitoring HW:
  - 18 Event Counters
  - 18 Counter Configuration Control Registers
    - Configure how to count
  - 45 Event Selection Control Registers
    - Configure what to count
  - Additional Control Registers

# Our Event-Counter: Performance Reader



- Performance Reader implemented as Linux Loadable Kernel Module
  - Implements 6 syscalls:
    - *select_events()*
    - *reset_event_counter()*
    - *start_event_counter()*
    - *stop_event_counter()*
    - *get_event_counts()*
    - *set_replay_MSRs()*

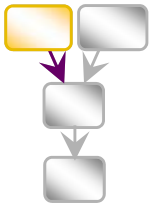
- User Level Interface:
  - Defines the events
    → Starts counters
  - Stops counters
    → Reads counters & TSC

```
Stopping Counter 17:

       Original CCCR[0X371]:      0X0000000000003B000
       Verified CCCR[0X371]:      0X0000000000003A000
result:3284

PERFORMANCE COUNTERS:
====================

   Counter          Value                 DIFF
   ---------      -----------           ---------

     0               778742                700832
     1           1098181002            1098159315
     2                    0                    0
     3                    0                    0
     4                    0                    0
     5                    0                    0
     6                    0                    0
     7                    0                    0
     8                    0                    0
     9                    0                    0
    10                    0                    0
    11                    0                    0
    12            932134310             932112394
    13          23025284483           23024933823
    14           8010600542            8010450663
    15                    0                    0
    16            932115216             932112394
    17           5004051815            5004037602
   TSC      3386216329638090           46331546804

  Delta Time: 33.0939620029 [s]


  Instr-ns Retired:          8.01045e+09
  Uops Retired:              2.30249e+10
  LOADs Retired:             5.00404e+09
  L1 Load Misses Retired:    9.32112e+08
  L2 HITS (shrd,Exc,Mod):    1.09816e+09

Our Load's L1 Hit Rate Approximation:    6.86397 %

L1 Hit Rate DERIVED FROM L2 ACCESSES:    -9.72732 %
```

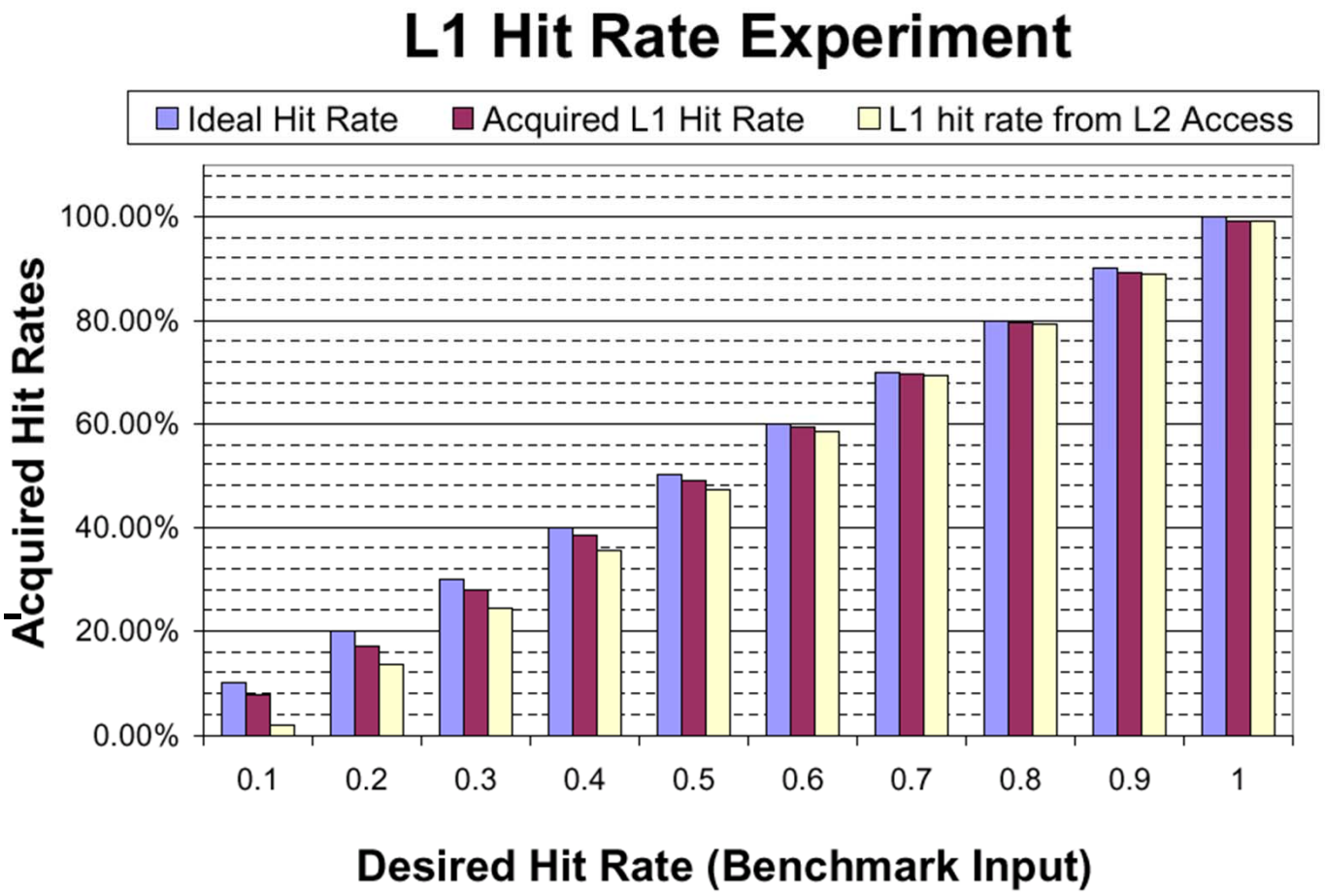
## Event Types:

- 59 event classes
- 100s of events to count

# Performance Reader: Example Validation

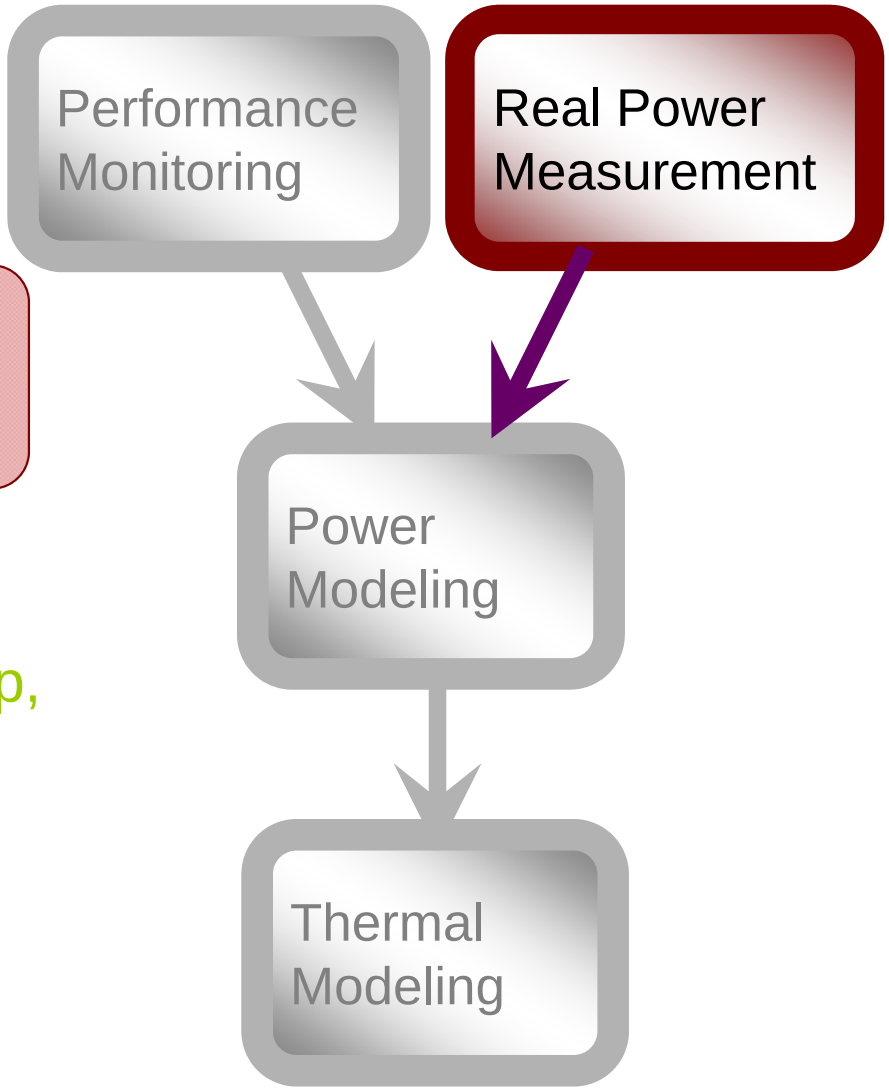


- L1_Dcache benchmark
- Controls cache hit behavior
- Validated against measured cache events
- Vary hit rate from 0-100%



# Processor Power Measurement

- Related Work
- Performance Monitoring
  - P4 Performance Counters
  - Performance Reader LKM
- Real Power Measurement
  - P4 Power Measurement Setup
  - Examples
- Power Modeling
  - P4 Power Model
  - Model + Measurement Sync Setup, Verification
- Thermal Modeling
  - Refined Thermal Model
  - Ex: Ppro Thermal Model



# P4 Power Measurement Setup

Clamp ammeter on 12V lines on measured CPU

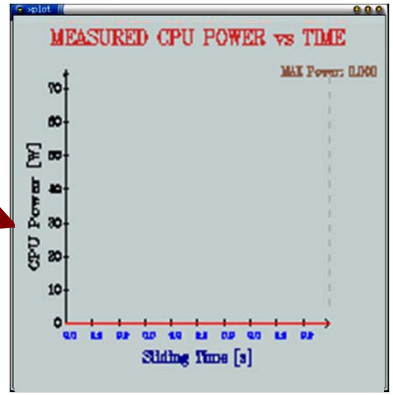
1mV/Adc conversion

DMM reading clamp voltages

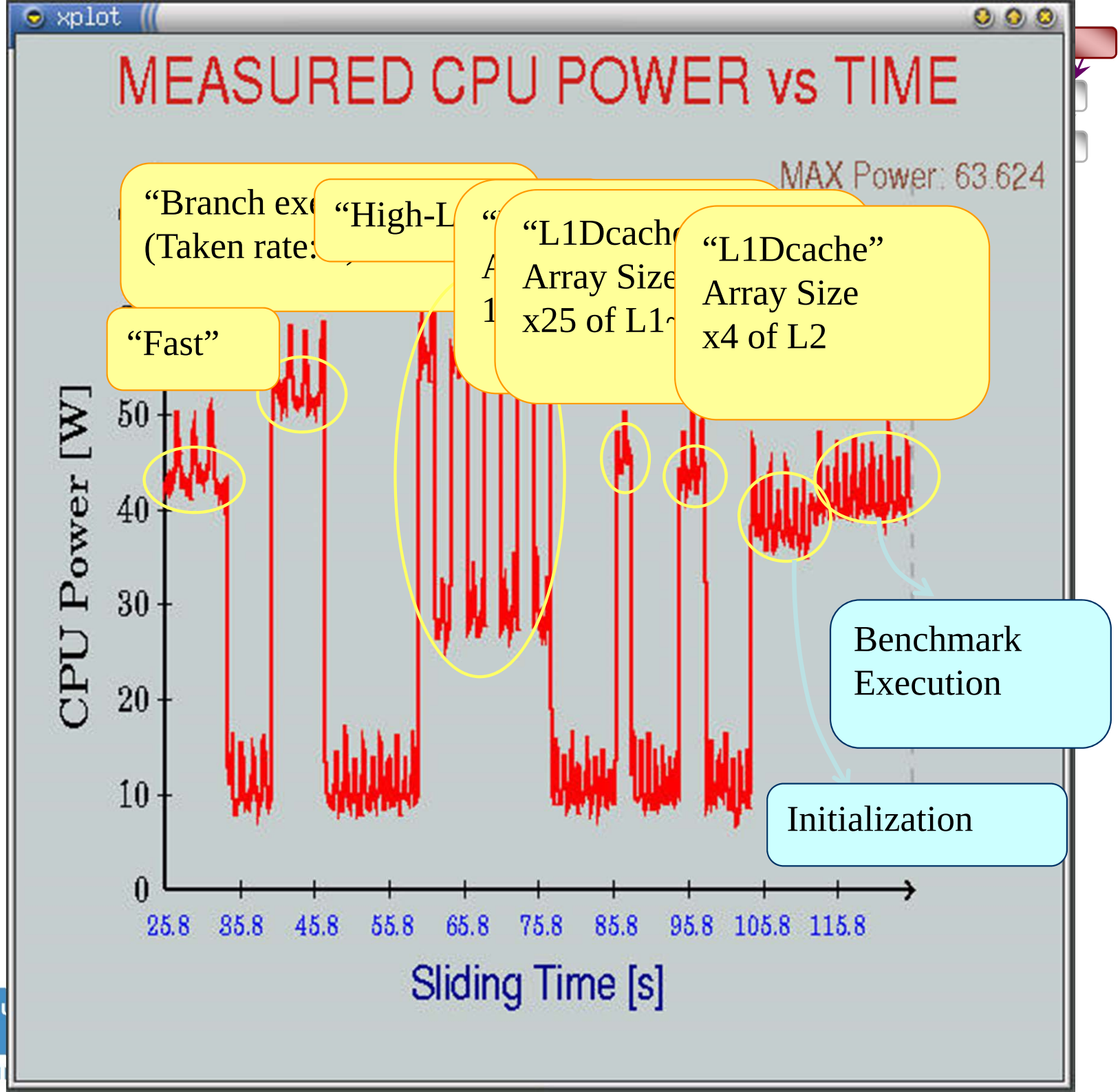
Voltage readings via RS232 to logging machine

Serial Reader
(PowerMeter)
(PowerPlotter)

Convert to Power vs. time window

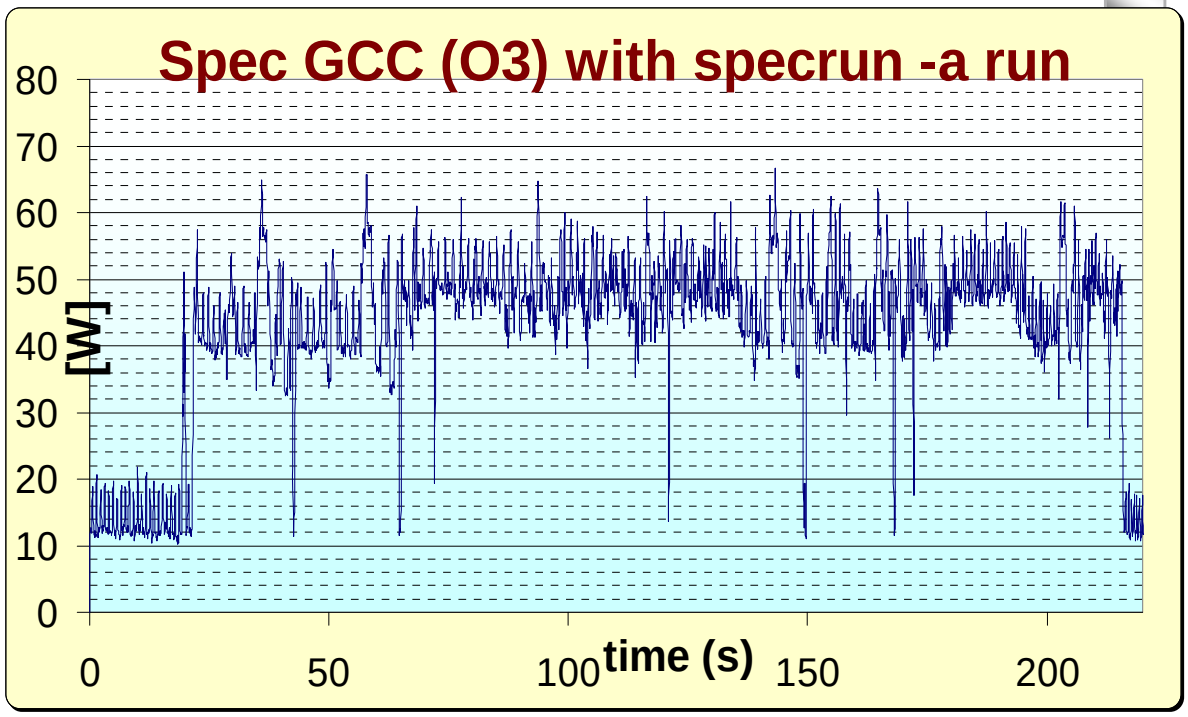


# PowerPlotter: Example

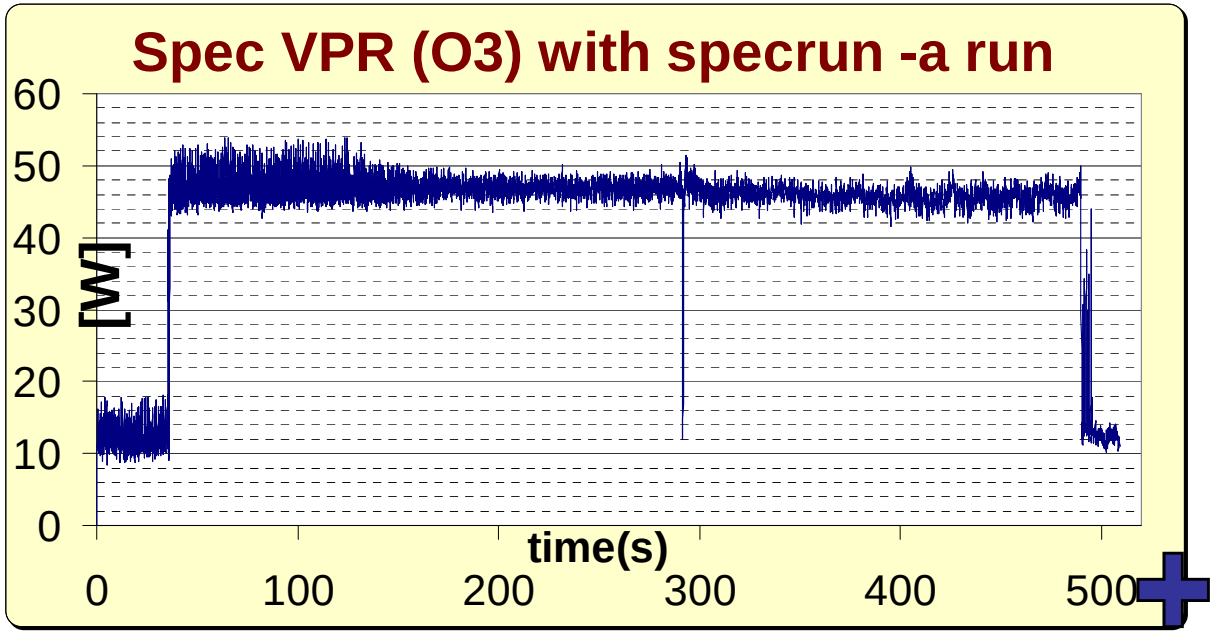


# SPEC Power Examples

- Different programs show very different power characteristics

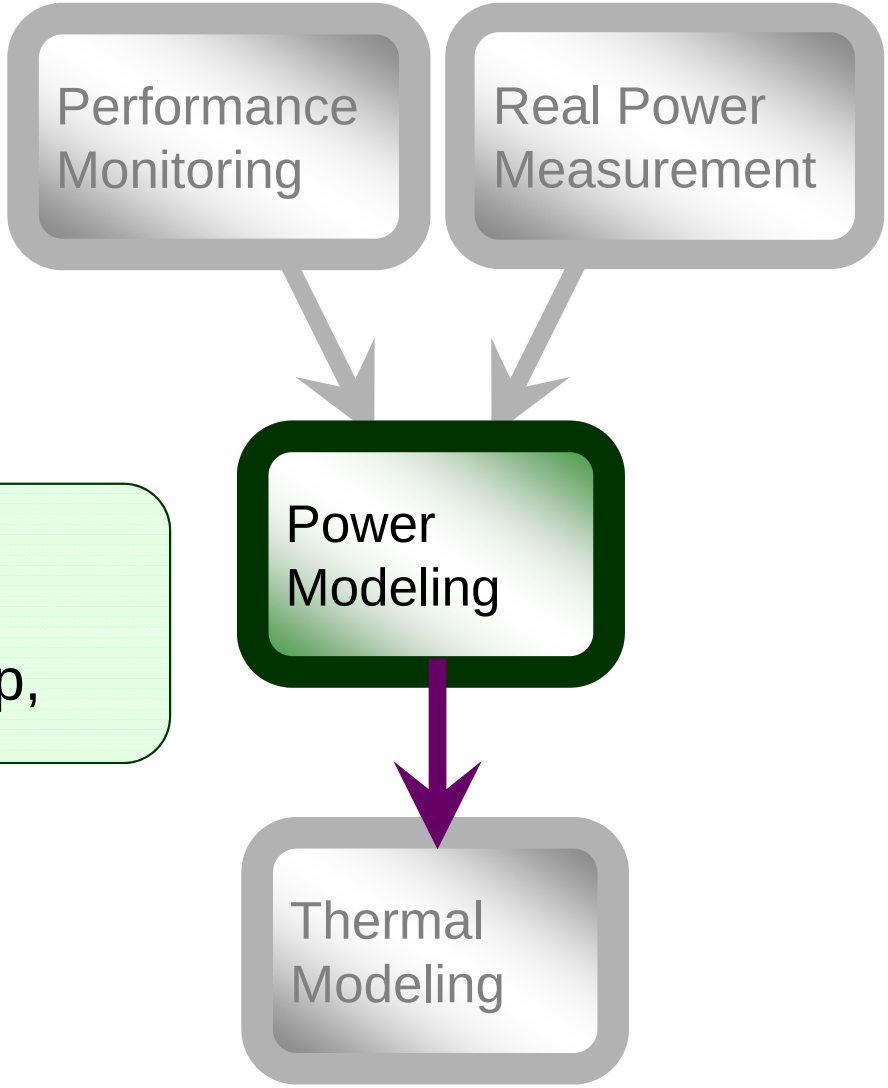



- Timescale of interest can be huge => inaccessible via simulation

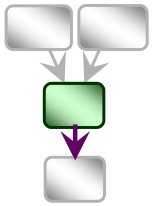


# Processor Power Modeling

- Related Work
- Performance Monitoring
  - P4 Performance Counters
  - Performance Reader LKM
- Real Power Measurement
  - P4 Power Measurement Setup
  - Examples
- Power Modeling
  - P4 Power Model
  - Model + Measurement Sync Setup, Verification
- Thermal Modeling
  - Refined Thermal Model
  - Ex: Ppro Thermal Model



# P4 POWER MODEL



Define components (I.e. L1 cache, BPU, Regs, etc.), whose powers we'll model:

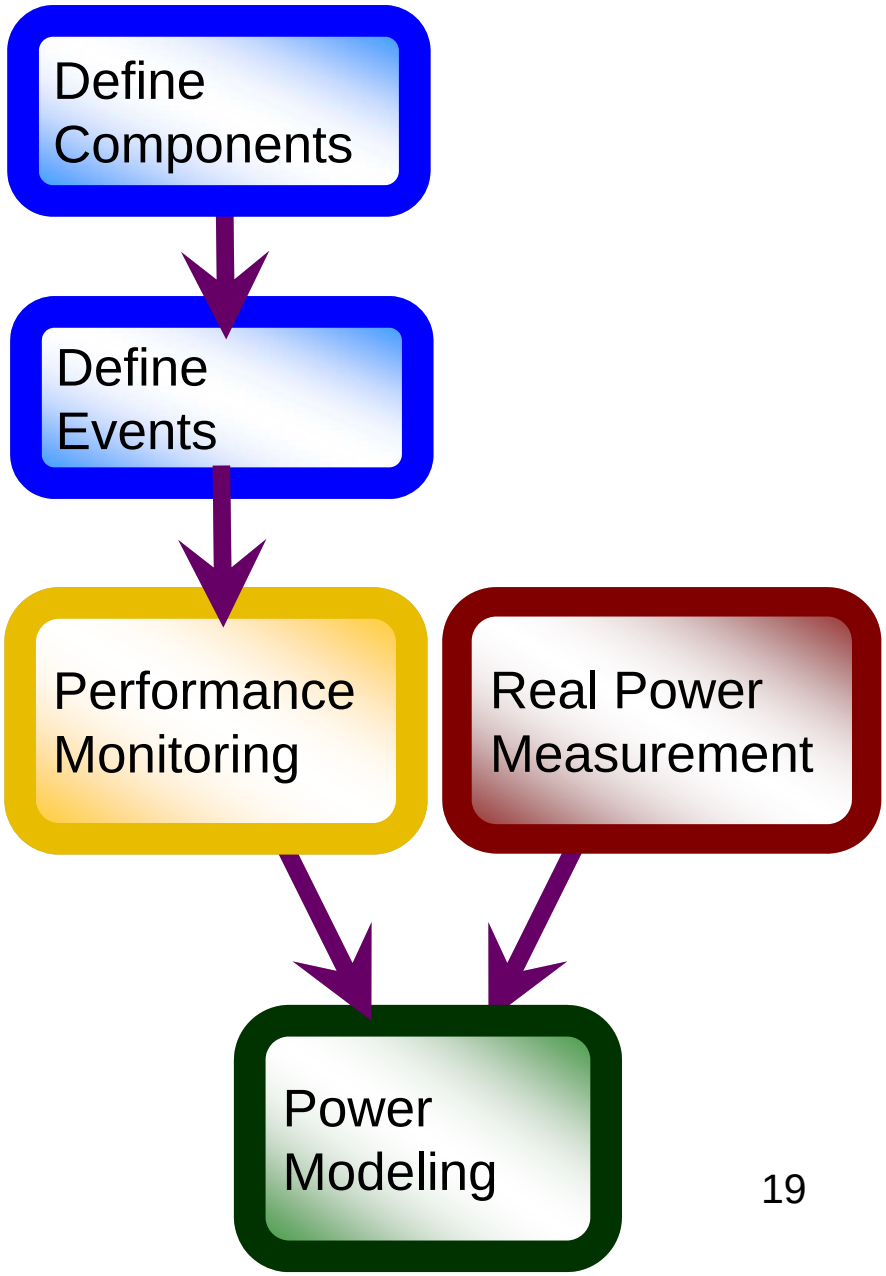
- from annotated layout

Determine combination of P4 events that represent component accesses best

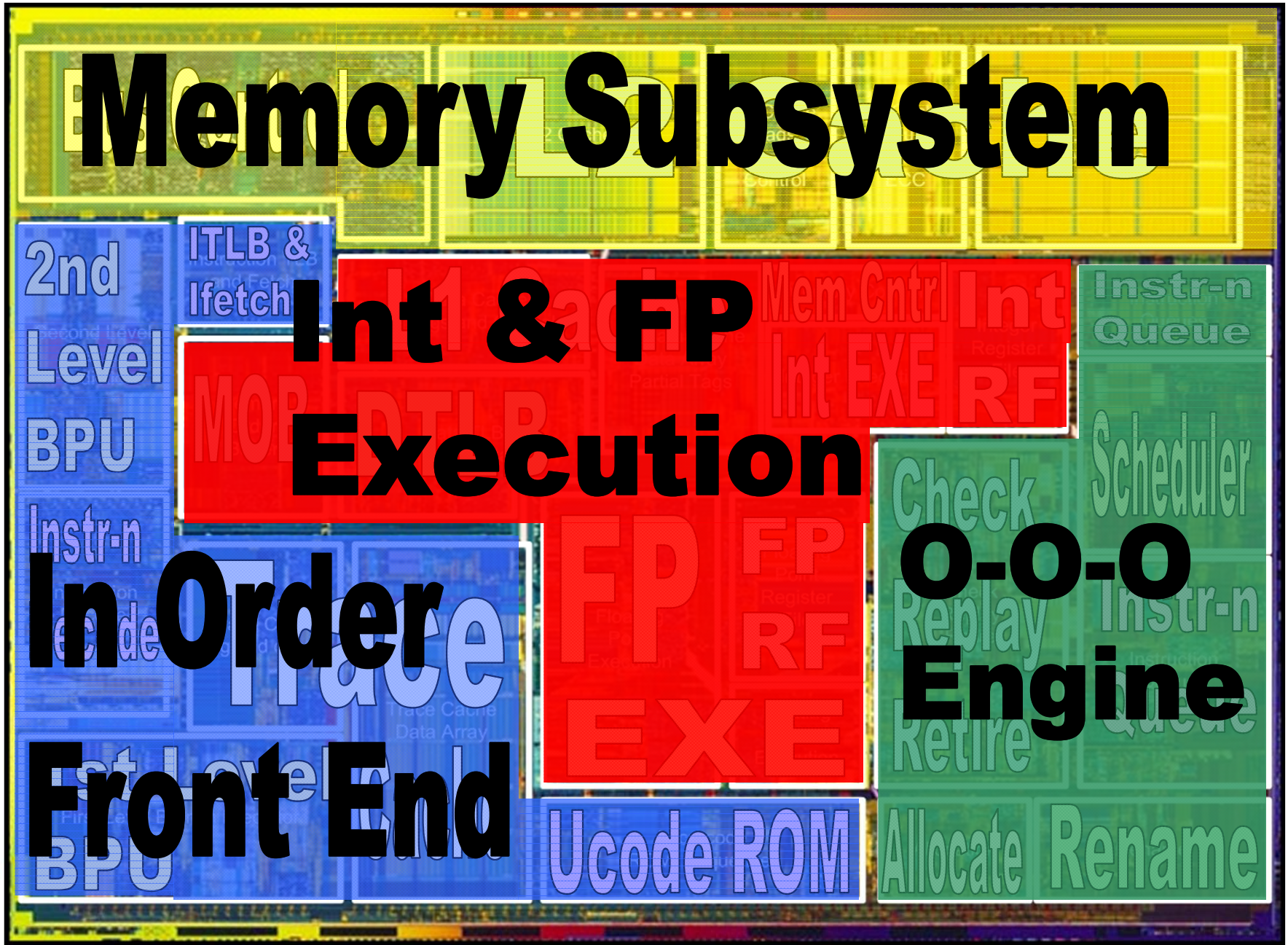
Gather counter info with minimal power overhead and program interruption

Convert counter info into component power breakdowns

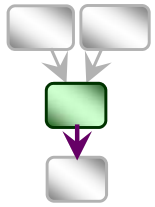
Verify total power against measured processor power



# Defining Components



# Defining Events → Access Rates

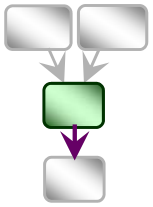


- We determined 24 events to approximate access rates for 22 components
- Used Several Heuristics to represent each access rate
- **Examples:**

| | Access Heuristics |
|---|---|
| Bus Control | $\frac{IOQ\ Allocation}{\Delta Cycles_1} + \frac{Bus\ Ratio \cdot FSB\ Data\ Activity}{\Delta Cycles_2}$ |
| Front End BPU | $\frac{8 \cdot ITLB\ Reference}{\Delta Cycles_1} + \frac{Branch\ Retired}{\Delta Cycles_2}$ |
| L1 Cache | $\frac{Ld\ Port\ Replay + St\ Port\ Replay}{\Delta Cycles_1} + \frac{Front\ End\ Event}{\Delta Cycles_2}$ |
| Trace Cache | $\frac{Uop\ Queue\ Writes}{\Delta Cycles_1}$ |
| Integer Execution | $2 \cdot (\frac{Uop\ Queue\ Writes}{\Delta Cycles_1} - FP\ Exe.\ Access\ Rate) -$ $L1\ Cache\ Access\ Rate - \frac{Branch\ Retired}{\Delta Cycles_2}$ |

- Need to rotate counters 4 times to collect all event data
  - Used 15 counters & 4 rotations to collect all event data

# Access Rates → Component Powers



- "Performance Counter based Access Rate estimations are used as proxy for max component power weighting together with microarchitectural details in order to estimate processor sub-unit powers"

$$Power(C_i) = AccessRate(C_i) \cdot ArchitecturalScaling(C_i) \cdot MaxPower(C_i) + NonGatedClockPower(C_i)$$

  - EX: Trace cache delivers 3 uops/cycle in deliver mode and 1 uop/cycle in build mode:
    - Power(TC)=[Access-Rate(TC)/3 + Access-Rate(ID)]
      x MaxPower(TC)
      + Non-gated TC CLK power
- Total power is computed as the sum of all 22 component powers + measured idle power (8W):

$$Total\ Power = \sum_{i=1}^{22} Power(C_i) + Idle\ Power$$

# Experiment Setup – Recall:

Clamp ammeter on 12V lines on measured CPU

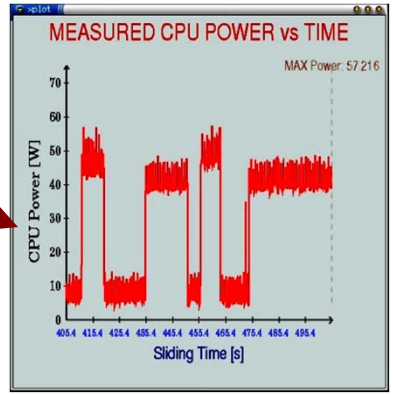
1mV/Adc conversion

DMM reading clamp voltages

Voltage readings via RS232 to logging machine

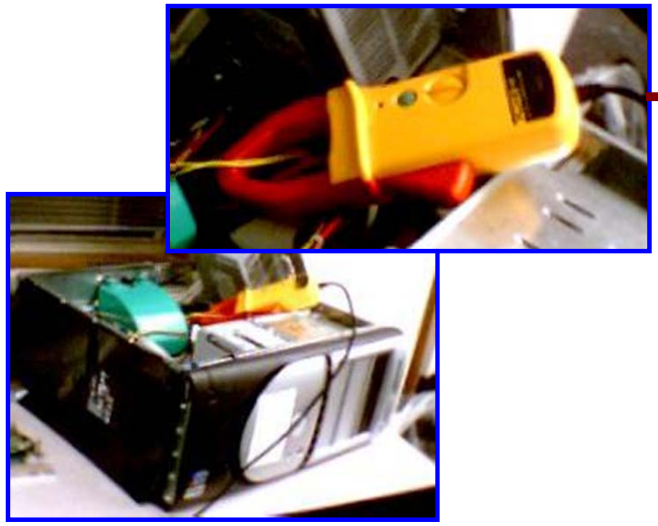
Serial Reader
(PowerMeter)
(PowerPlotter)

Convert to Power vs. time window




Departament d'Arquitectura de Computadors
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# Experiment Setup



1mV/Adc conversion

Voltage readings via RS232 to logging machine

# Experiment Setup

POWER SERVER

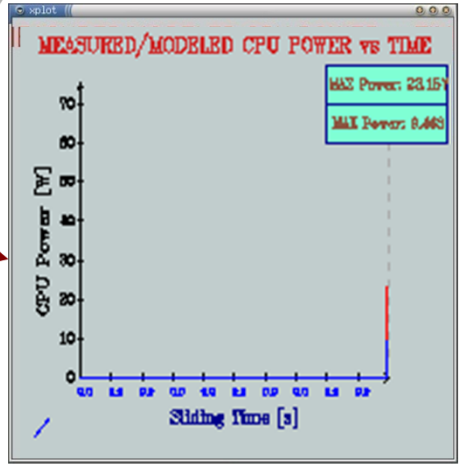
1mV/Adc conversion

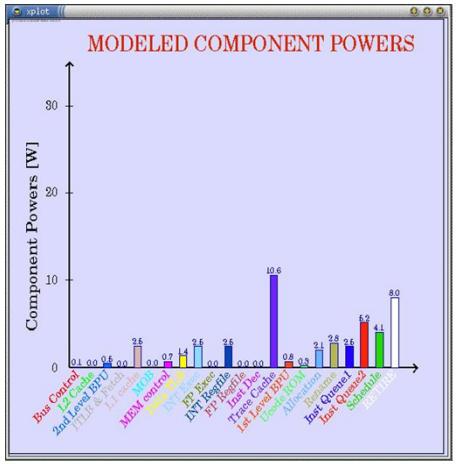
Voltage readings via RS232 to logging machine

Component access rates over ethernet

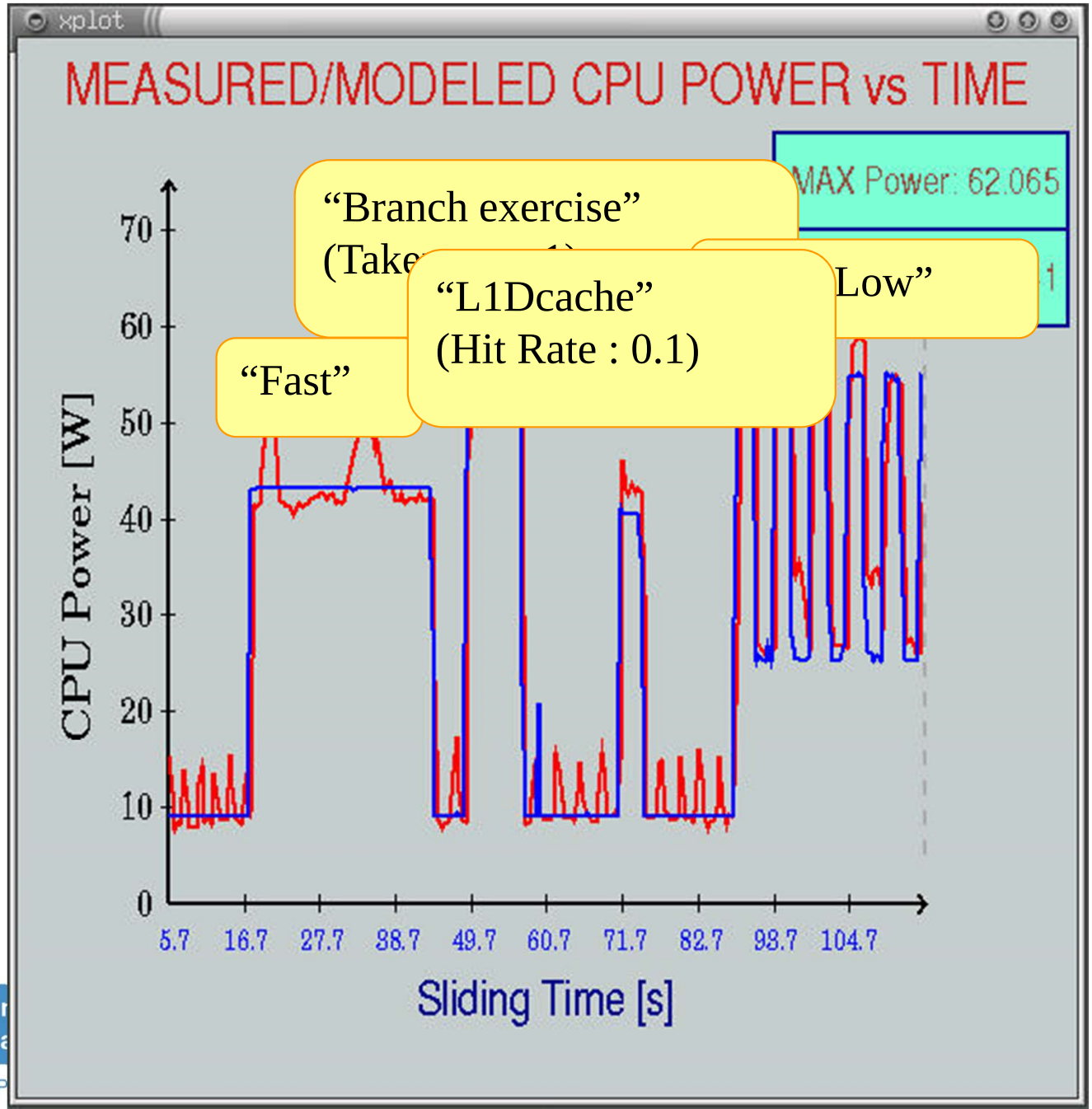
POWER CLIENT

Convert voltage to measured power
Convert access rates to modeled powers
Sync together in time window



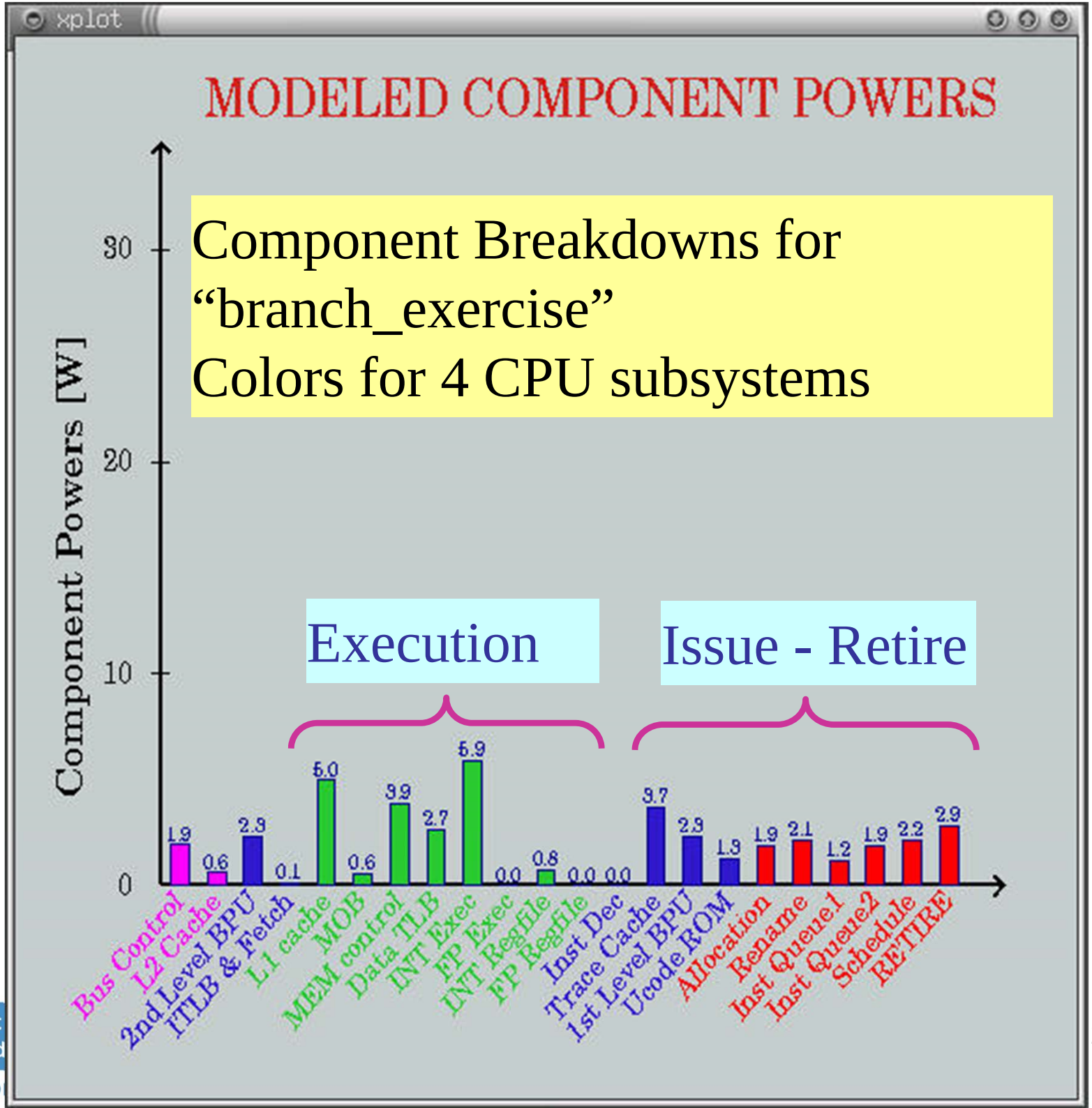
# Tuning Benchmarks




Measured

Modeled

# Component Breakdowns



# Benchmark Power Breakdowns

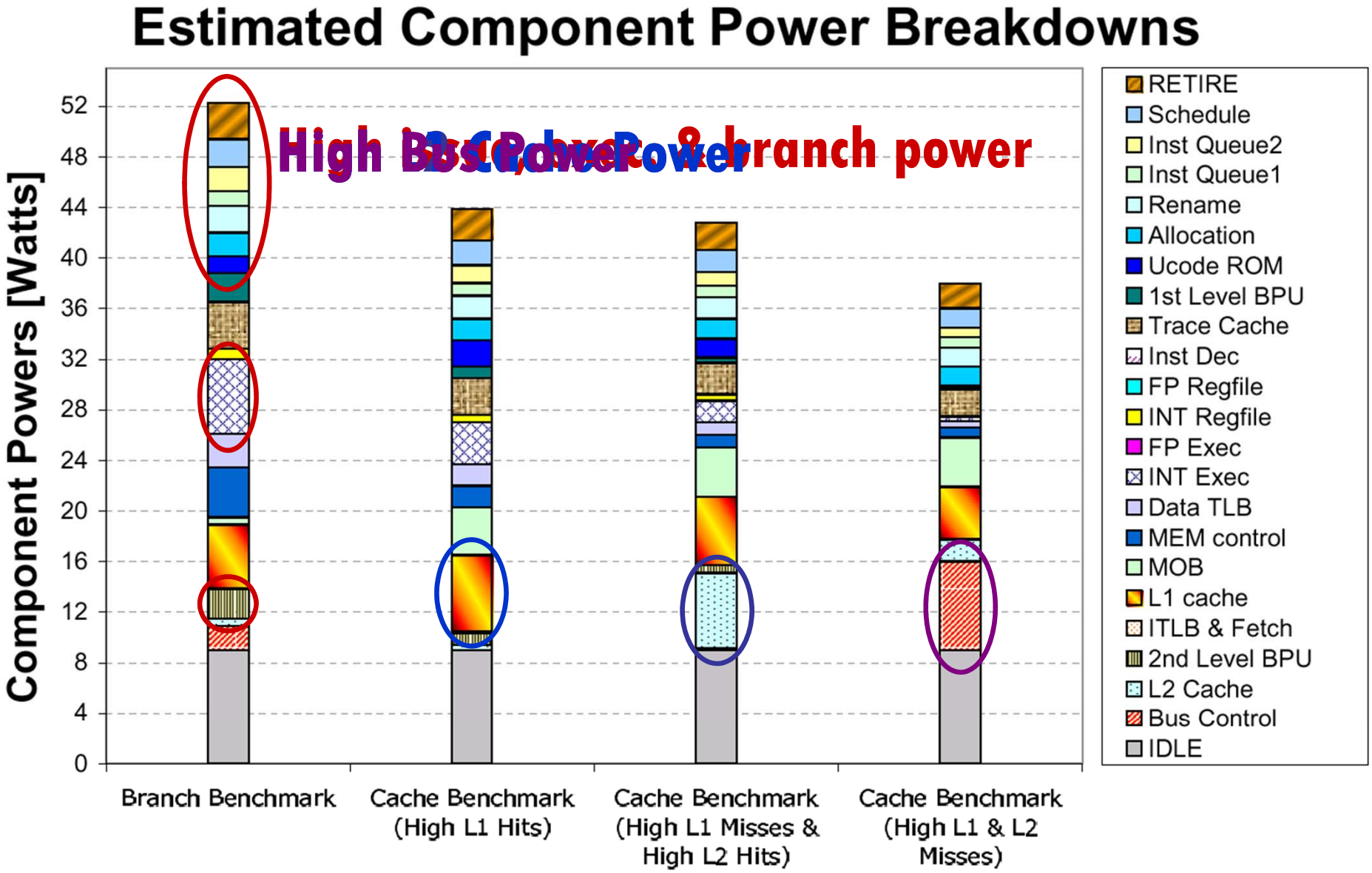
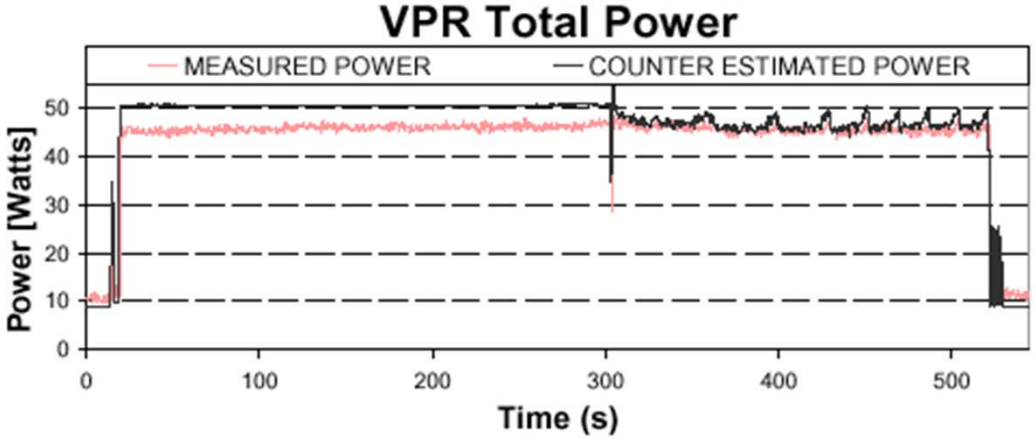
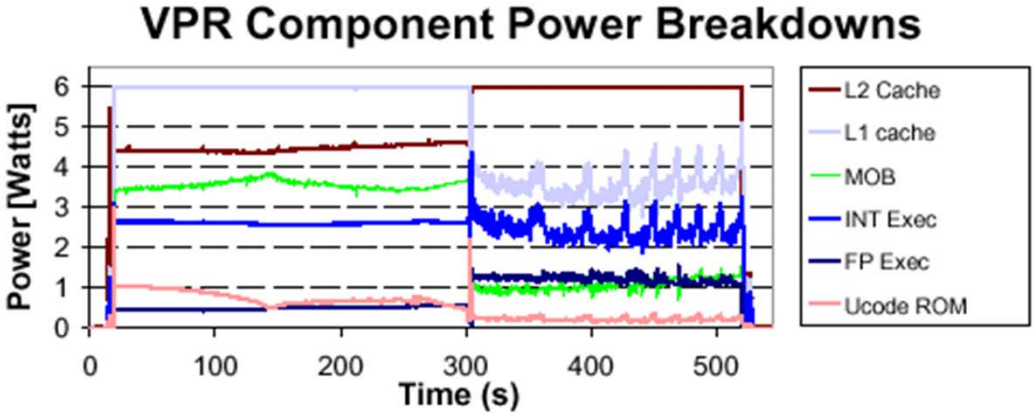



**Figure 8. Power breakdowns for branch and cache benchmarks.**

# SPEC2000 Results

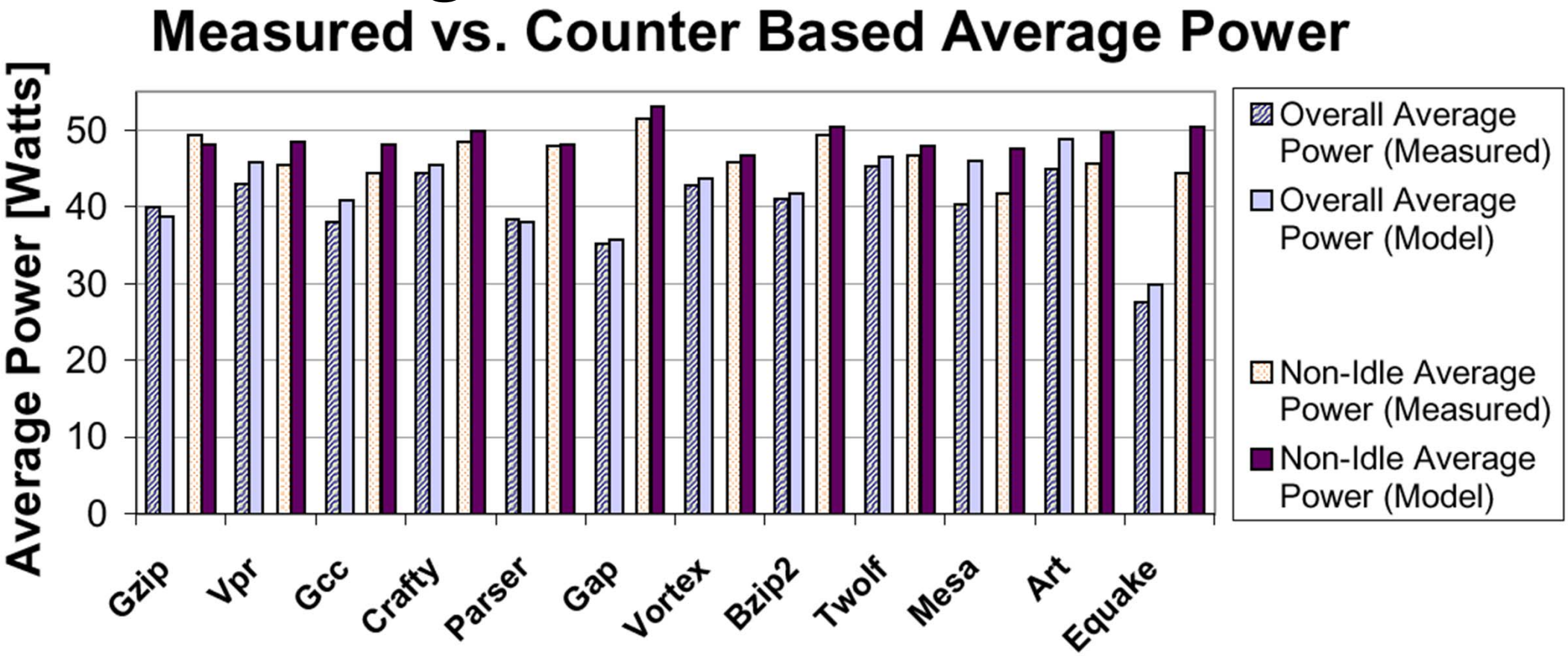



Equake Elaboration: (FP benchmark)
Initialization and computation phases
FP intensive mesh computation phase
Initialization with high complex IA32 instructions

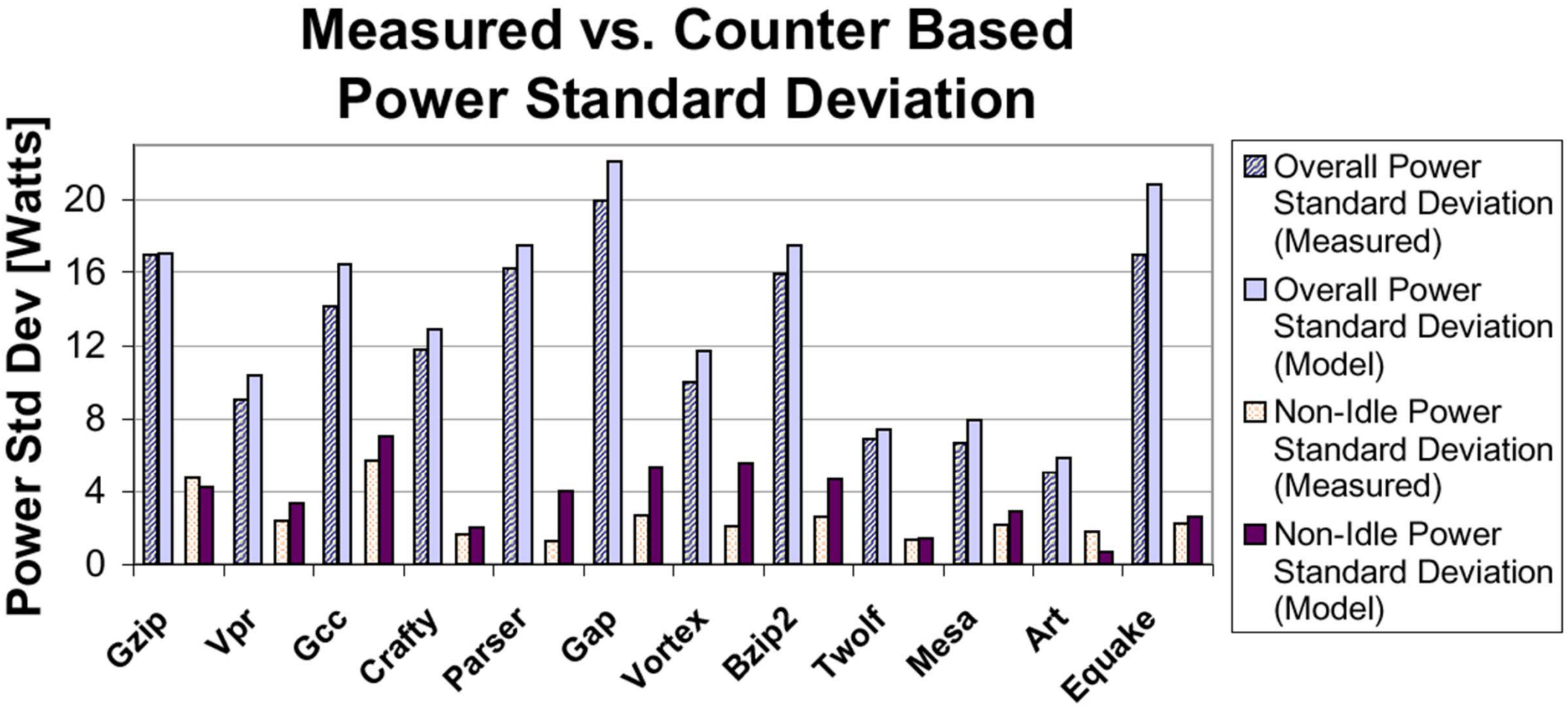
Twolf Elaboration:(Integer benchmark)
Several loop computations traversing memory
<High Memory Power>
Although ~const. Total power, component powers have slight gradients

# Average SPEC Total Powers




- 1st set: Overall, 2nd set: Non-idle power
- Average difference between measurement and estimation: 3W
- Worst case: Equake (5.8W)

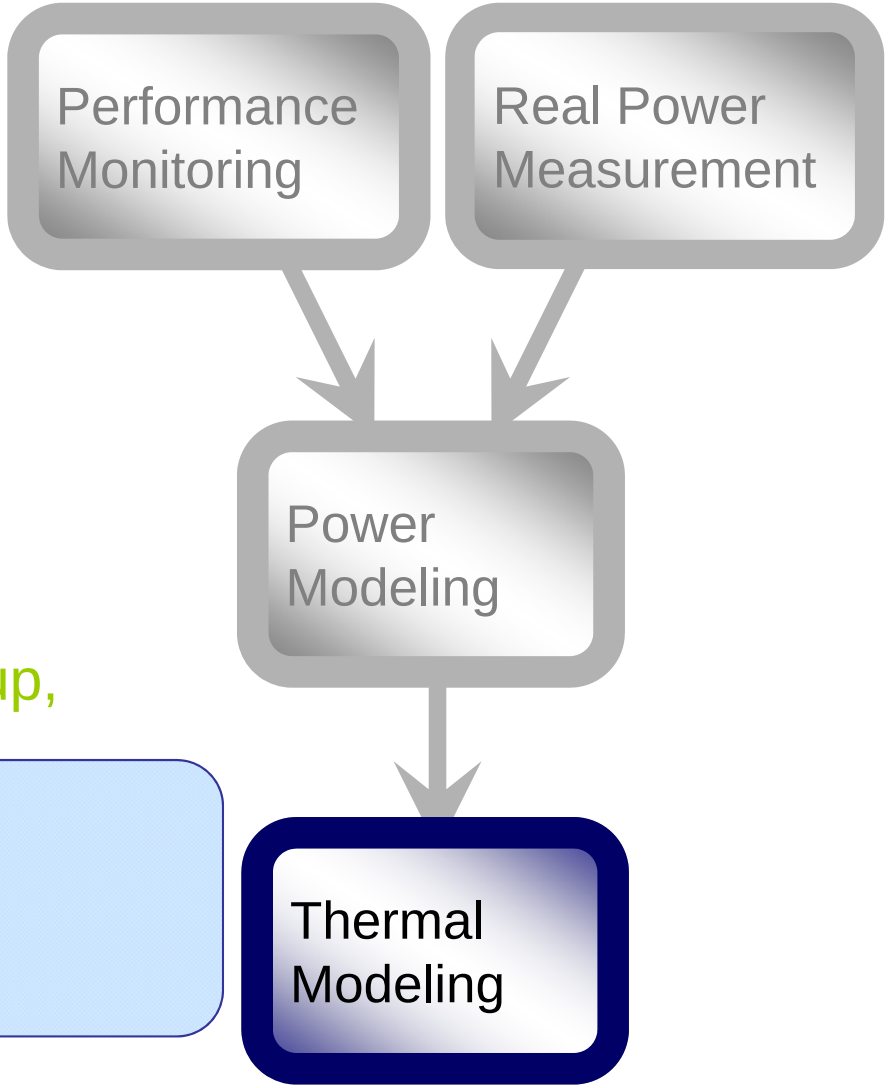
# Stdev of SPEC Total Powers




- 1st set: Overall, 2nd set: Non-idle power
- Average difference: 2W
- Worst case: Vortex (3.5W)

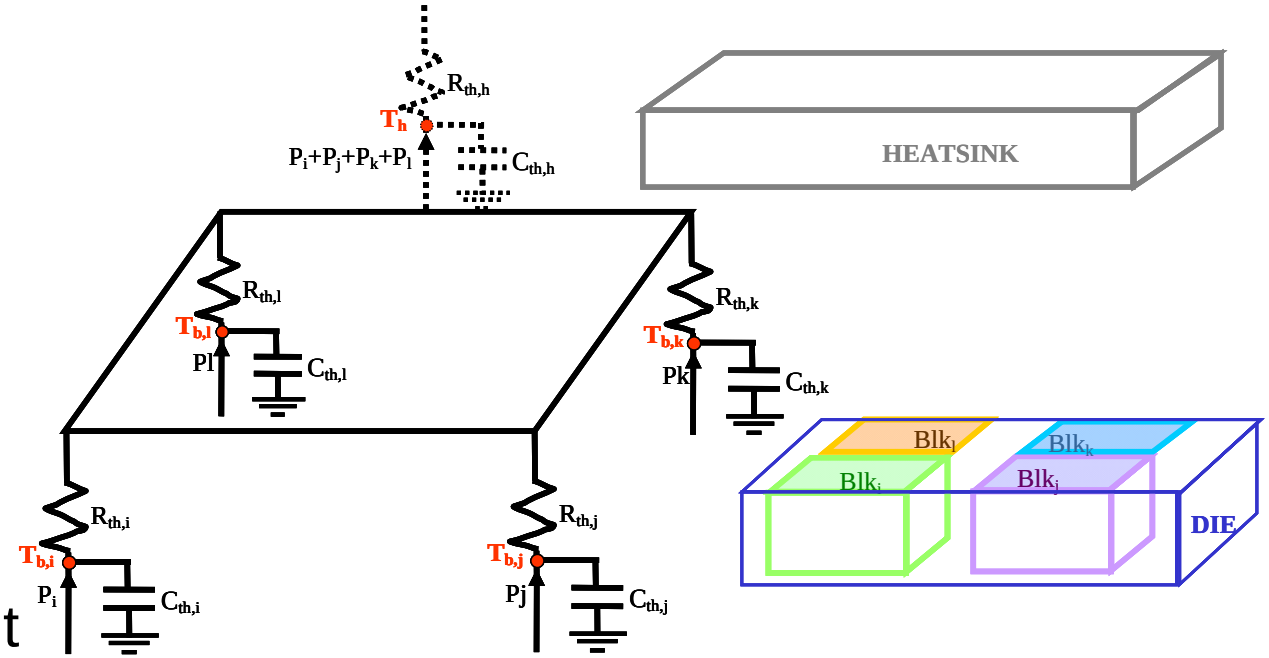
# Thermal Model

- Related Work
- Performance Monitoring
  - P4 Performance Counters
  - Performance Reader LKM
- Real Power Measurement
  - P4 Power Measurement Setup
  - Examples
- Power Modeling
  - P4 Power Model
  - Model + Measurement Sync Setup, Verification
- Thermal Modeling
  - Brief Thermal Model Intro
  - Ppro Thermal Model Results

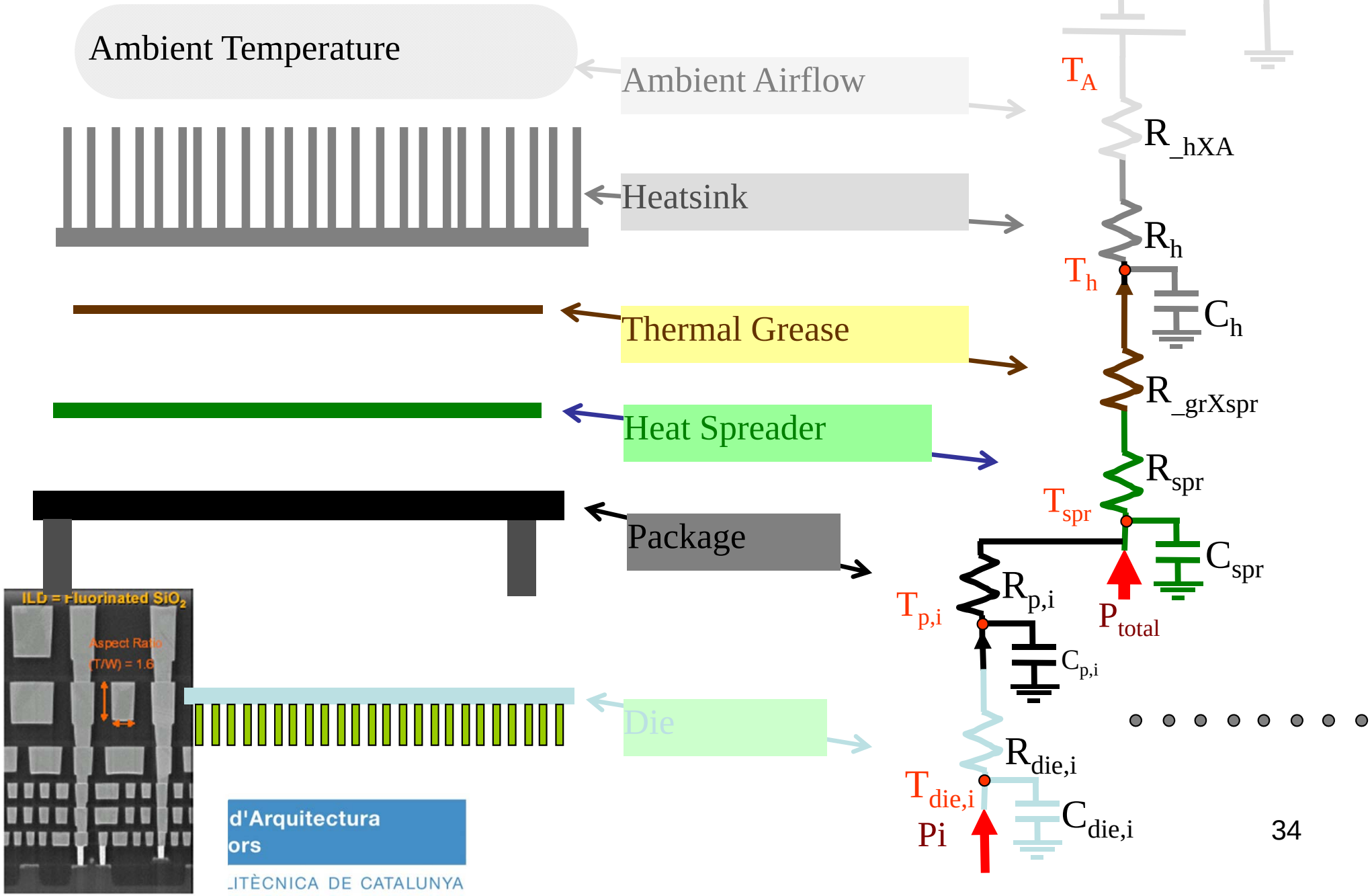


# THERMAL MODELING: A Basic Model

- Based on lumped R-C model from packaging
- Built upon power modeling
  - Sampled Component Powers
  - Respective component areas
  - Physical processor Parameters
    - Packaging
    - Heat Transfer

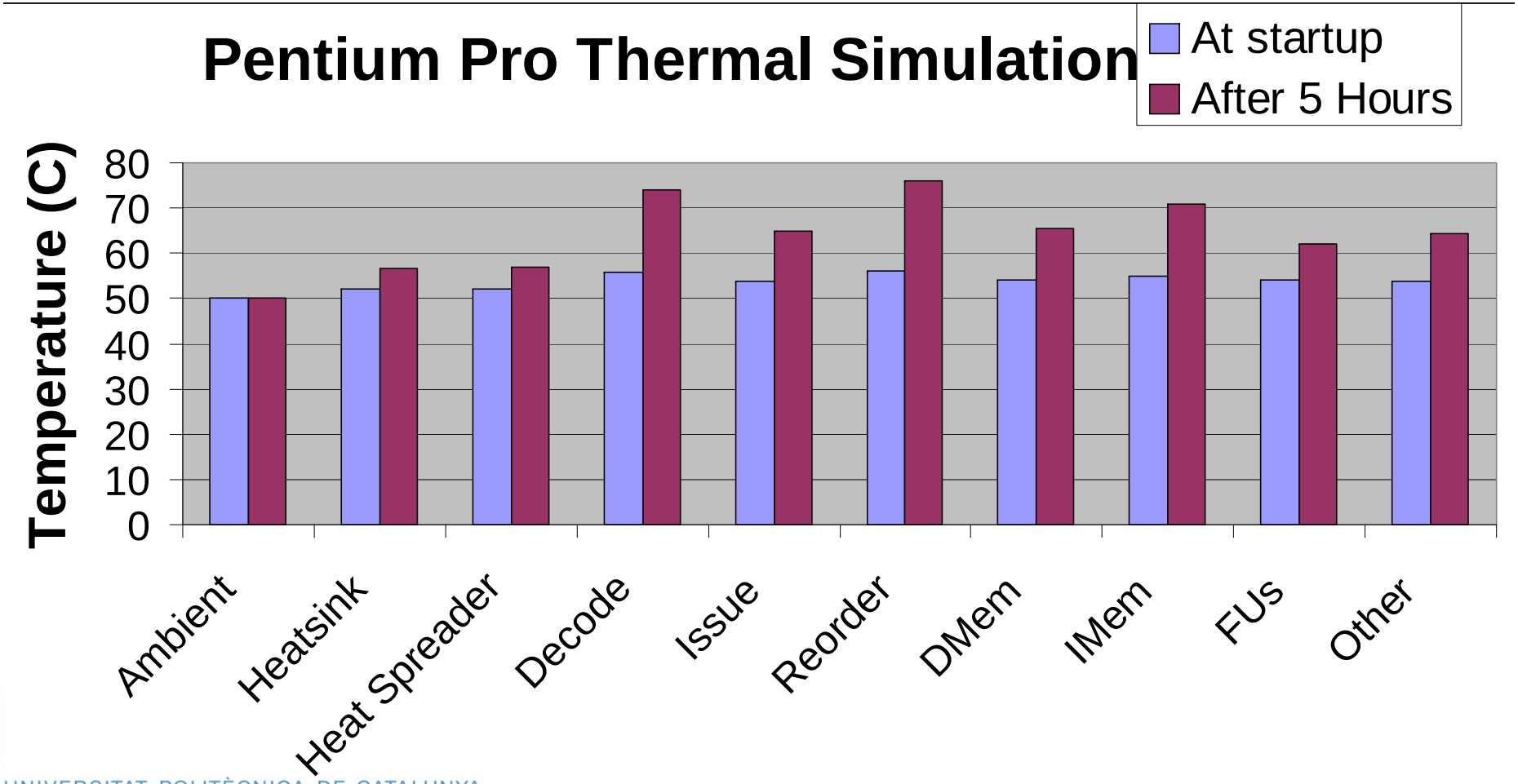


# Physical Structure vs. Thermal Model



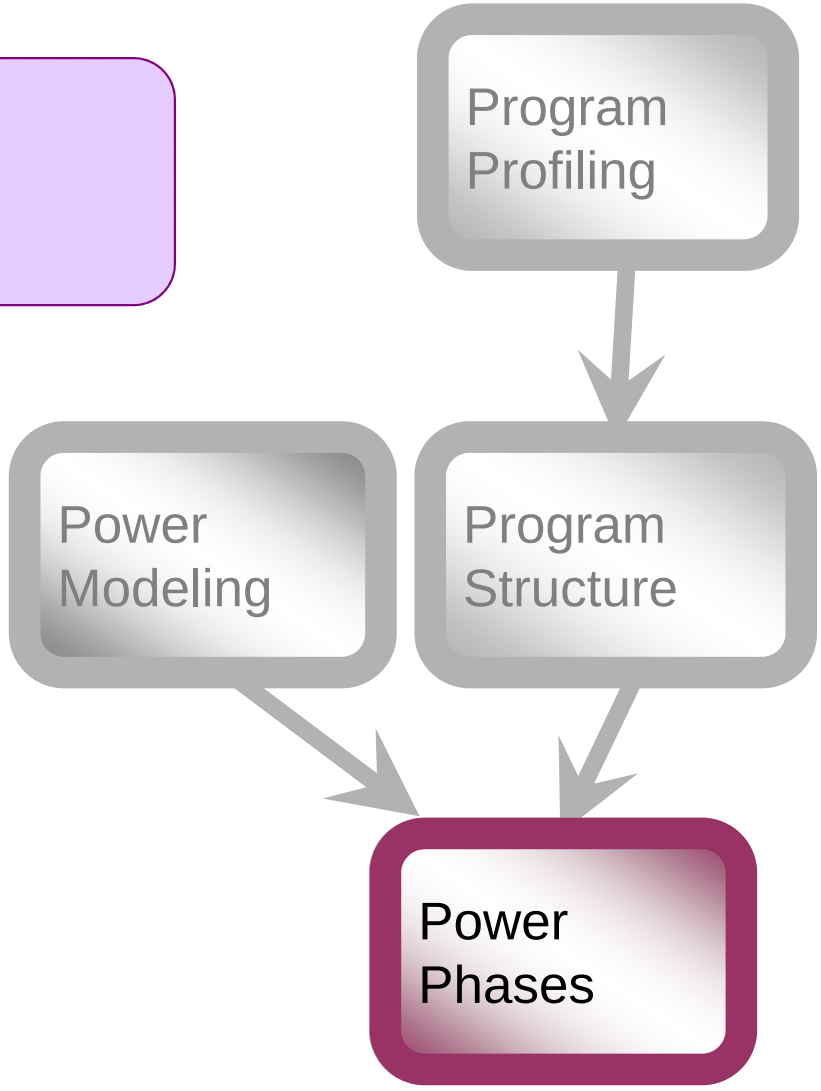
# Simulation Outputs

- Thermal nodes updated every $\Delta t$~20ms
  - Component Temperatures Build up to ~350K in ~5hrs
  - $T_{heatsink}$ moves very slowly as expected



# Power Phase Behavior

- Power Phase Behavior
  - Similarity Based on Power Vectors
  - Identifying similar program regions
- Profiling Execution Flow
  - Sampling process' execution
  - "PCsampler" LKM
- Program Structure
  - Execution vs. Code space
  - Power Phases ⇔ Exec. Phases
    - <OR VICE VERSA>



# Power Vectors for Similarity

- Similar to basic block vectors
- Use component power vector samples to represent program phases
- Consider Manhattan distance between 2 vectors as the ‘measure of dissimilarity’ between the corresponding execution points
- Construct a similarity matrix to represent similarity among all pairs of execution points
  - Each entry in the similarity matrix:

$$Similarity\ Matrix(r,c) = \sum_{i=1}^{22} |Power_r(C_i) - Power_c(C_i)|$$

# Gcc & Gzip Similarity Matrices

## Gzip Elaboration:

Much regular power behavior

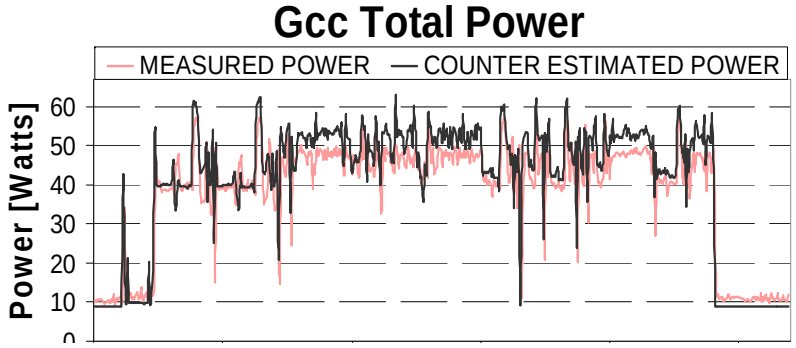
Spurious similarities such as 100-150s and 200-280 are distinguished by the similarity analysis

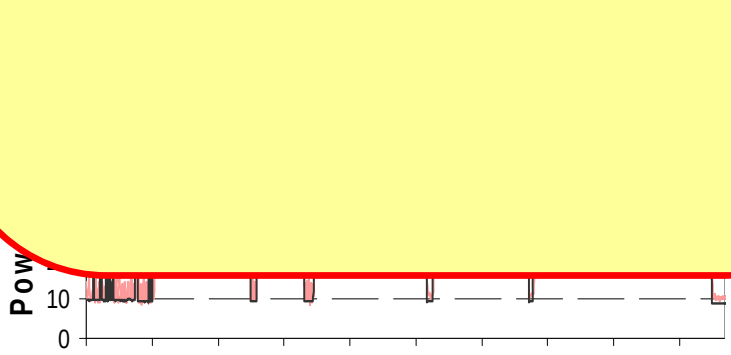
## Gcc Elaboration:

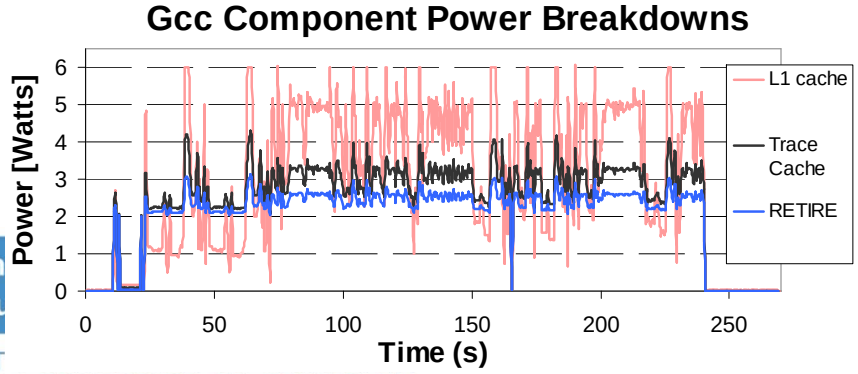
Very variant power

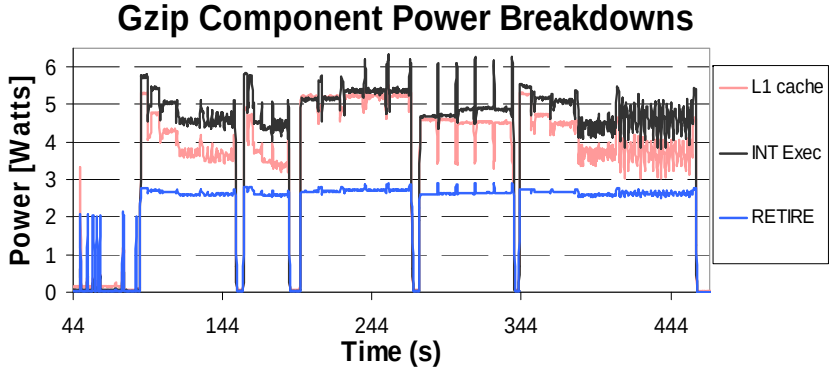
Almost identical power behavior at 30, 50, 180s.

Although 88s, 110s, 140s, 210s and 230s show similar total power; 88, 210 and 230 share higher similarity.



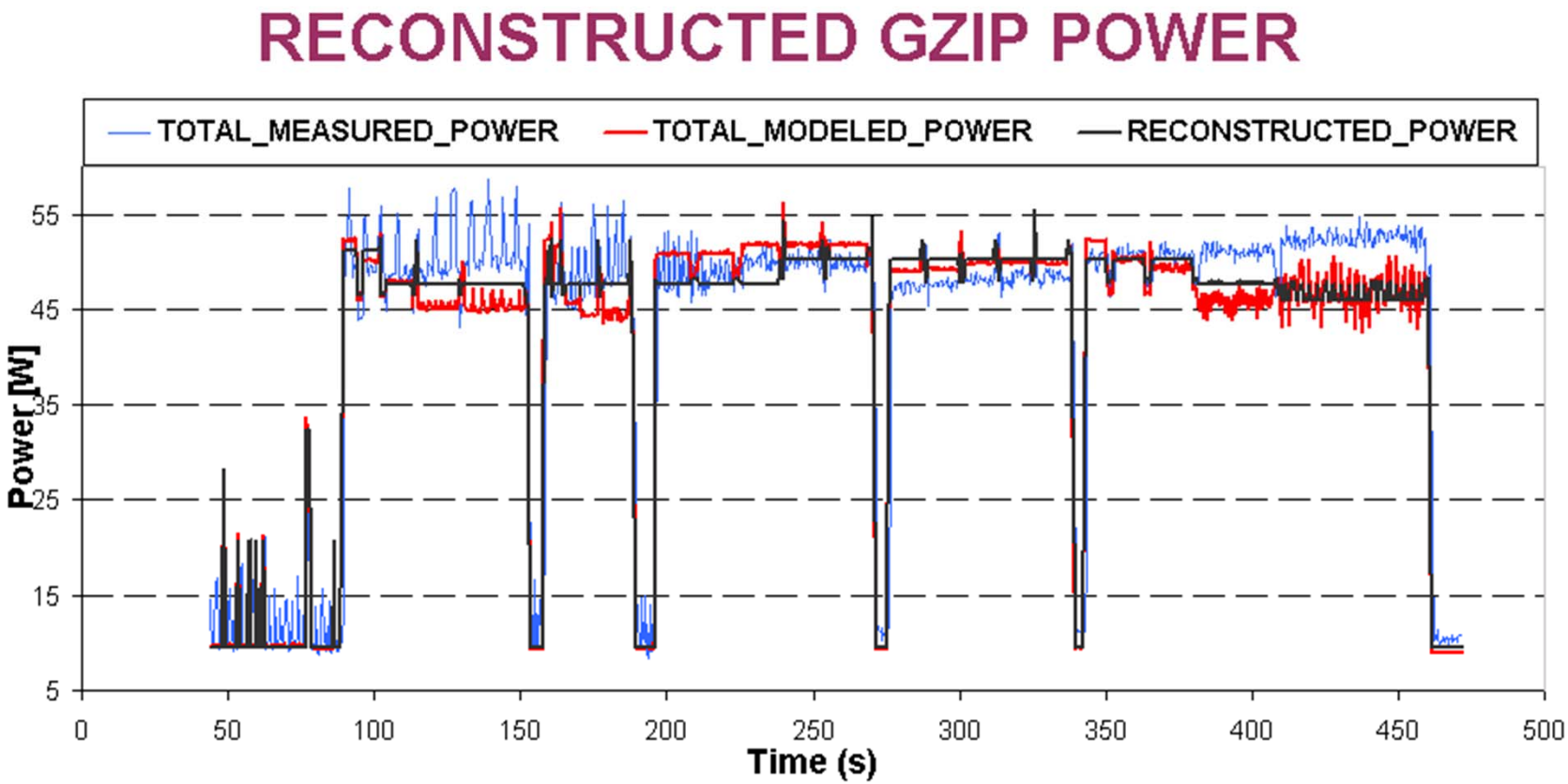

UPC

Departam de Compu

UNIVERSITAT

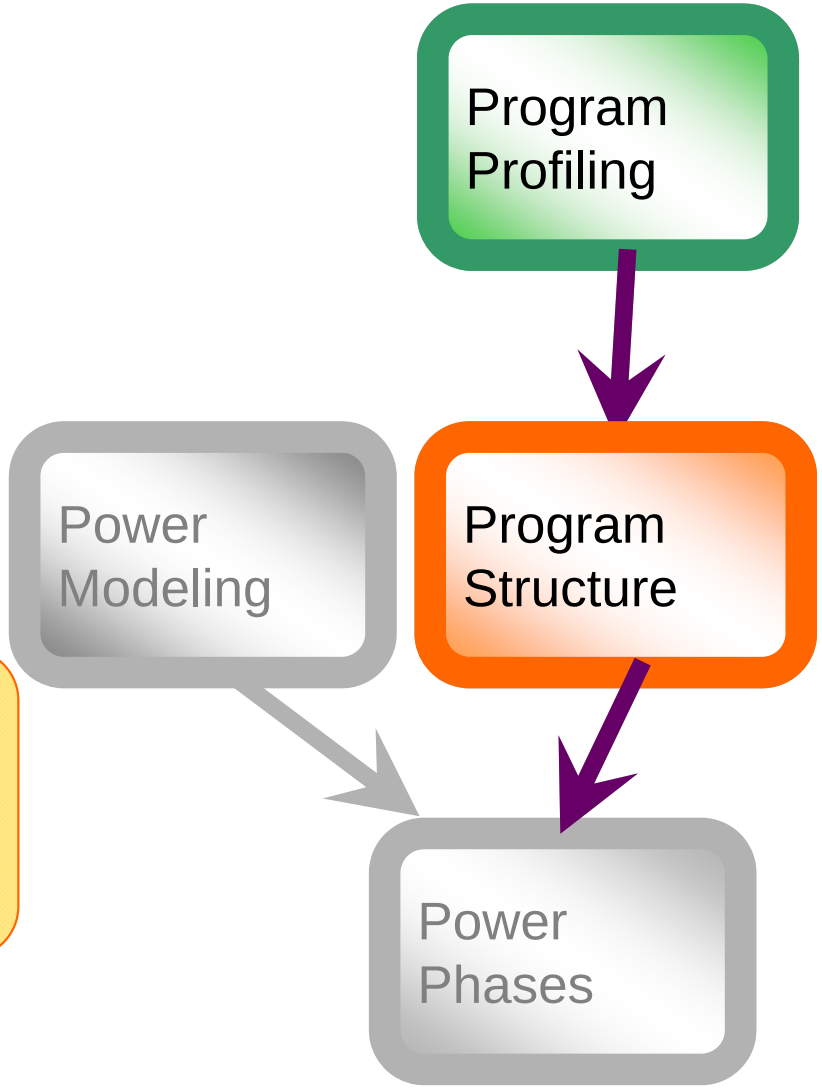
# Generating representative vectors

- Gzip has ~1000 power vectors
- Cluster vectors based on similarity
- Could we represent power behavior with reasonable accuracy, with a small number of ‘signature’ vectors?
- **Ex: 26 representative vectors with “Thresholding Algorithm”:**



# Program Execution Profile

- Power Phase Behavior
  - Similarity Based on Power Vectors
  - Identifying similar program regions
- Profiling Execution Flow
  - Sampling process' execution
  - "PCsampler" LKM
- Program Structure
  - Execution vs. Code space
  - Power Phases ⇔ Exec. Phases
    - <OR VICE VERSA>



# Program Execution Profile

- Sample program flow simultaneously with power
  - LKM implementation: “PCsampler”
- Generate code space similarity in parallel with power space similarity
- Relative comparisons for:
  - Complexity
  - Accuracy
  - Applicability, etc.

# Round-up

- Hw counter give a precise measure of the CPU performance
- Good heuristics are needed to turn these event counters to activities
- Fast and accurate performance/power characterization
- Limited to that specific CPU. No design-space exploration possible.

# Agenda

- Introduction
- Using Hardware Counters
  - Based on PARAPET group work (Princeton)
- Architecture Simulation
- Statistical sampling
- Related Work
- Conclusions

# Temperature

- Typical temperatures of Pentium 4 processor
  - P-N junction temperature = up to 73 °C
  - Die temperature = 100 °C
- Current (& thus speed) decreases exponentially
  - Diodes: $I = I_s(e^{V/nVt} - 1)$, $Vt = kT/q$, T=temperature
  - Resistors: $R \propto T$, $I = V/R$
  - Rule of thumb: Speed $\downarrow$ 0.15% per °C
- But leakage current increases with temperature
  - Thermal runaway
  - $\uparrow$ temp $\rightarrow$ $\uparrow$ leakage $\rightarrow$ $\uparrow$ self-heating $\rightarrow$ $\uparrow$ temp
  - Motivates multiple Vt (see 0.1 μm paper by Taur)
- Reliability decreases exponentially
  - T $\uparrow$ 10-15°C $\rightarrow$ Chip lifetime $\downarrow$ 50%!
- Physical warping & cracking
  - Different CTE at package material interfaces

# Effects of High Temperature

- Typical temperatures of Pentium 4 processor
  - P-N junction temperature = up to 73 °C
  - Die temperature = 100 °C
- Current (& thus speed) decreases exponentially
  - Diodes: $I = I_s(e^{V/nVt} - 1)$, $Vt = kT/q$, T=temperature
  - Resistors: $R \propto T$, $I = V/R$
  - Rule of thumb: Speed ↓ 0.15% per °C
- But leakage current increases with temperature
  - Thermal runaway
  - ↑ temp → ↑ leakage → ↑ self-heating → ↑ temp
  - Motivates multiple Vt (see 0.1 μm paper by Taur)
- Reliability decreases exponentially
  - T ↑ 10-15°C → Chip lifetime ↓ 50%!
- Physical warping & cracking
  - Different CTE at package material interfaces

# Effects of High Current

- Voltage drop
  - IR drop due to high current flowing through power grid
- di/dt current transients
  - Clock edges
  - Powering up/down large logic blocks
  - Add decoupling capacitors on chip and package to mitigate
- Electromigration
  - Metal atoms of thin wires physically move over time
  - Eventually, short circuits & open circuits which break the chip

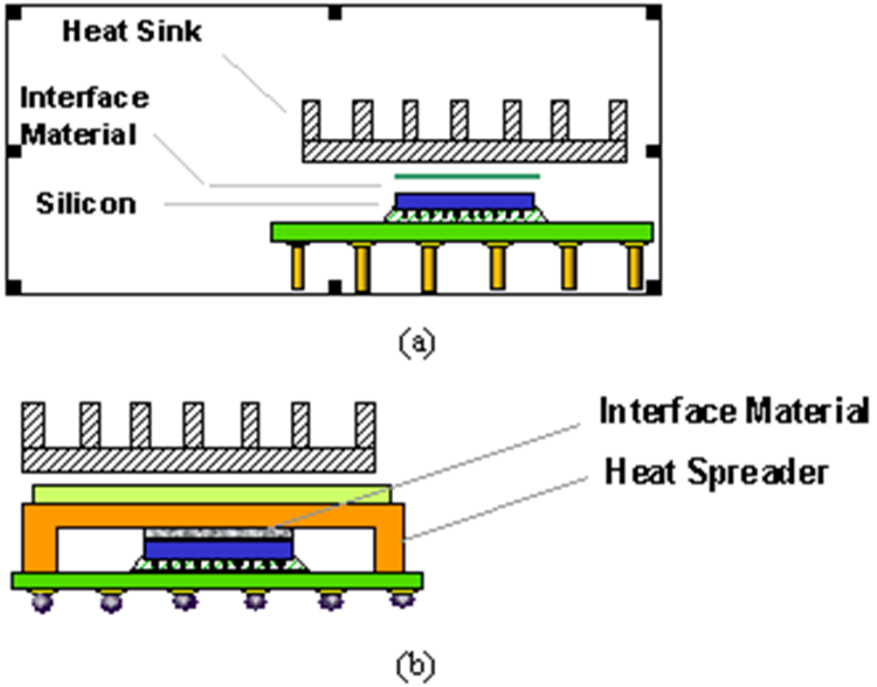
# Temperature During Test

- Burn-in testing
  - Test at high Vcc and high temperature
  - Goal: Induce "infant mortality" failures without breaking good, reliable specimens
- Test 1,000 chips at once
  - Burn-in "oven" is now a refrigerator!
  - Need to dissipate up to 50 kW!
- Emerging tester technology: Hydraulic cooling

"Thermal Challenges during Microprocessor Testing", Intel Technology Journal, Q3 2000

# Thermal Solutions

- Heat sink
  - Mounted on processor package
- Passive cooling
  - Remote system fan
- Active cooling
  - Fan mounted on sink
- Heat spreaders
  - Increase surface area
  - Example:  Metal plate under laptop keyboard

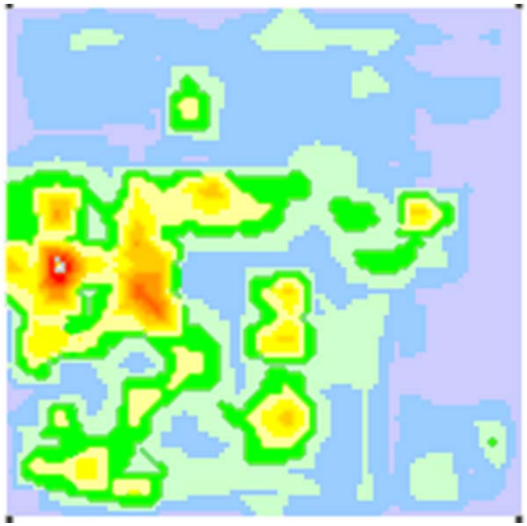



a. Heat sink mounting for low-power chip
b. Package design for high-power chips

"Thermal Challenges during Microprocessor Testing", Intel Technology Journal, Q3 2000

# Estimating Power

- Architecture level
  - Statistical event simulation
  - Average cost of high-level operations
- RTL level
  - Effective frequency of gated clocks
  - Validated for timing, logic, deadlock-free
- Floorplan level
  - Create thermal image of die
  - Overlay floorplan to identify hottest functional blocks
- Schematic level
  - Based on circuit style (static, dynamic, etc.)



"Thermal Challenges during Microprocessor Testing", Intel Technology Journal, Q3 2000

"Managing the Impact of Increasing Microprocessor Power Consumption", Intel Technology Journal, Q1 2001

# Power Management on Pentium 4 Processor

- Over 400 power-saving features!
  - 20% of features = 75% of saved power
- Clock throttling
  - Thermal diode temperature sensor
  - Stop clock for a few microseconds
  - Output pin can be used by system to trigger other responses
- SpeedStep technology for mobile processors
  - Switch to lower frequency and voltage
  - Depends on whether power source is battery or AC
  - Can be manually overridden by Windows control panel

"Managing the Impact of Increasing Microprocessor Power Consumption", Intel Technology Journal, Q1 2001

# Pentium 4 Processor Multi-level Powerdown

- Level 0 = Normal operation (includes thermal throttle)
- Level 1 = Halt instructions (less processor activity)
- Level 2 = Stop Clock (internal clocks turn off)
- Level 3 = Deep sleep (remove chip input clock)
- Level 4 = Deeper sleep (lower Vdd by 66%)
  - For “extended periods of processor inactivity”
  - QuickStart technology – resume normal operation from Deeper Sleep
- Note: We haven’t even talked about system powerdown modes, like removing power from processor, stopping hard disks, dimming or turning off the display…

“Managing the Impact of Increasing Microprocessor Power Consumption”, Intel Technology Journal, Q1 2001

# RELATED WORK

- **Implementing counter readers:**
  - PCL [Berrendorf 1998], Intel VTune, Brink & Abyss [Sprunt 2002]
- **Using counters for Performance:**
  - HPC [Crummey 2001], CPU profilers
- **Using counters for Power:**
  - CASTLE [Joseph 2001], power profilers
  - event driven OS/cruise control [Bellosa 2000,2002]
- **Real Power Measurement:**
  - Compiler Optimizations [Seng 2003]
  - Cycle-accurate measurement with switch caps [Chang 2002]

# RELATED WORK

- **Power Management and Modeling Support:**
  - Instruction level energy [Tiwari 1994]
  - PowerScope: Procedure level energy [Flinn 1999]
  - Event counter driven energy coprocessor [Haid 2003]
  - Power-breakdown driven energy reduction [Huang 2001]
  - Virtual Energy Counters for Mem. [Kadayif 2001]
  - ECOsystem: OS energy accounting [Ellis 2002]

- **Thermal Management and Modeling Support:**
  - PID based DTM [Skadron 2002]
  - Architectural Thermal Model [Skadron 2003]
  - Evaluating DTM techniques [Brooks 2001]

# RELATED WORK – performance monitoring

- **<u>implementing counter readers:</u>**

- PCL Performance Counter Library, by Rudolf Berrendorf (University of Applied Sciences Bonn-Rhein-Sieg), Heinz Ziegler, and Bernd Mohr at the Central Institute for Applied Mathematics (ZAM) at the Research Centre Juelich , Germany
  - uniform interface for several architectures (intel Pentium,MMX, Pro, III, 4/linux; IBM Power3, Power3-II/AIX; etc.)
  - Software library with C, C++, Java & Fortran Bindings
  - Kernel patch (Mikael Pettersson)→ recompile

- PAPI Performance Application Programming Interface Project, by Jack Dongarra, Kevin London, Shirley Moore, Philip Mucci, etc., at Innovative Computing Lab, CS dept., University of Tennessee
  - Standard Simple high level API and low level programmable interface
  - Supports Pentium, MMX, Pro, III/Linux, Windows; Power 3,4/AIX; etc.
  - PerfCtr kernel patch (Mikael Pettersson) → recompile

# RELATED WORK – performance monitoring

- **implementing counter readers:**
- Perfmon Performance Monitoring Tool by Richard Enbody, Associate Professor Department of Computer Science and Engineering, Michigan State University.
  - For SUN Ultra-Sparc & Ppro
  - Device Driver (LKM)
- Rabbit Performance Counters Library by Don Heller, Scalable Computing Laboratory, Iowa State University
  - for Intel Pentium MMX, Pro, II, III/Linux; AMD/Linux
  - functions to access from within C
    - Cleanest of all, but still ~30 files & ~50instructions
  - LKM
- Intel's VTune Performance analyzer
  - Windows & Linux <New>
- IBM's HPM toolkit
  - Power 3,4/AIX
- Brink and Abyss Pentium 4 Performance Counter Tools For Linux, by Brinkley Sprunt, Electrical Engineering, Bucknell University
  - brink: high level perl script to read experiment/config files
  - abyss: c program to access counters
  - abyss_dev: device driver for counter access
  - EBS kernel patches: to handle PMIs

# RELATED WORK – performance monitoring

- **using counter readers:**
- CASTLE Project by Margaret Martonosi and Russ Joseph, Princeton University
  - acquire Ppro counter data to model component power breakdowns
- Frank Bellosa, “Benefits of Event Driven energy Accounting in Power Sensitive Systems”, 9th SIGOPS European workshop, 2000
  - Counters to show power ~ k x instr-ns/cycle (PII)
  - OS power optimizations:
    - Throttle down CPU/extend thread time for cache hit/slow down CPU core if main memory is accessed
- Andreas Weissel, Frank Bellosa, “Process Cruise Control: Event driven clock scaling for dynamic power management”, CASES 2002
  - Use event counters info to scale individual thread frequencies
  - Intel Xscale / Modified Linux kernel

# RELATED WORK – performance monitoring

- **using counter readers:**
- HPC Toolkit, by John Mellor-Crummey, Rob Fowler, CS Dept. Rice University
  - Uses perf counter data for profiling
  - converts raw profiling information into platform independent XML formats and produces performance metric correlations from multiple sources
  - Used in compiler optimizations
- Jennifer Anderson, et al, “Continuous Profiling: Where Have All the Cycles Gone?”, ACM Transactions on Computer Systems, Vol. 15, No. 4, November 1997, pp. 357 - 390.
  - Performance analysis example – from DEC
  - Data collection by counter sampling, performance info from program level to individual instructions

# RELATED WORK – real power

- CASTLE Project by Margaret Martonosi and Russ Joseph, Princeton University
  - Shunt R over Ppro power lines to measure total processor power
- John Seng, Dean Tullsen, “Effect of compiler optimizations on Pentium 4 Power consumption”, 7th Annual Workshop on Interaction between Compilers and Computer Architectures, February, 2003
  - Shunt R between VRM and CPU
- Marc A. Viredaz, Deborah A. Wallach, “Power Evaluation of Itsy Version 2.3”, tech. note TN-57, WRL, Compaq Computer Corp., 2000
  - similar series R to estimate battery life of itsy pocket computer

# RELATED WORK – real power

- Frank Bellosa, “Benefits of Event Driven energy Accounting in Power Sensitive Systems”, 9th SIGOPS European workshop, 2000
  - Crude Current measurement with DMM for Pentium II to help define per instruction powers
- Andreas Weissel, Frank Bellosa, “Process Cruise Control: Event driven clock scaling for dynamic power management”, CASES 2002
  - series sense resistor added to Intel IQ 80310 evaluation platform power supply, to measure energy effect of frequency scaling
- Naehyuck Chang, Kwanho Kim, and Hyun Gyu Lee, "Cycle-Accurate Energy Consumption Measurement and Analysis: Case Study of ARM7TDMI" ISLPED 2000 & IEEE Transactions on VLSI Systems, Vol. 10, pp. 146 - 154, Apr., 2002.
  - cycle accurate energy consumption measurement based on charge transfer
  - Inserts switch caps between power supply and Processor that switch with the same clock frequency!!

# RELATED WORK – power model

- **Simulation Tools:**

- WATTCH, by David Brooks and Margaret Martonosi, Princeton University, ISCA 2000
  - Architectural power simulator
  - Power Models intergrated upon SimpleScalar
- SimplePower by W. Ye, N. Vijaykrishnan, M. Kandemir, Penn-State University, and M. Irwin “The Design and Use of SimplePower: A cycle-accurate energy estimation tool”, DAC, June 2000
  - Execution driven, Cycle accurate, RTL power estimation
  - Emulates 5 stage pipe with SimpleScalar’s Integer ISA

# RELATED WORK – power model

- **Power Modeling:**

- R. Joseph and M. Martonosi. “Run-Time Power Estimation in High Performance Microprocessors”, International Symposium on Low Power Electronics and Design, 2001
  - complete CASTLE Project: Collects Ppro counter data and models component power breakdowns verifying against measured total power
  - Also Wattch simulation vs. counter approximation for SimpleScalar architecture

- Russ Joseph, David Brooks, and Margaret Martonosi, "Live, Runtime Power Measurements as a Foundation for Evaluating Power/Performance Tradeoffs" Workshop on Complexity Effectice Design (WCED, held in conjunction with ISCA-28), 2001
  - Evaluate power vs. performance by measuring total power and acquiring performance data from counters – i.e. Cache hit rate, branch prediction, bitline activity

# RELATED WORK – power model

- H. Zeng, X. Fan, C. Ellis, A. Lebeck, and A. Vahdat, “ECOSystem: Managing Energy as a First Class Operating System Resource”, Proceedings of ASPLOS X, Oct. 2002
  - Uses Currentcy Model (Fixed Power & Time budget for a task) for OS level energy management for battery life
  - ECOsystem is the Linux OS implementation <No counters>
  - Considers CPU ON/OFF → could do better with Power model
- H. Zeng, C. Ellis, A. Lebeck, A. Vahdat , “Currentcy: Unifying Policies for Resource Management”, USENIX 2003 Annual Technical Conference
  - Detailed description of currency (OS scheduling, etc.)
- Flinn J., Satyanarayanan, M., “PowerScope: A Tool for Profiling the Energy Usage of Mobile Applications”, Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications February, 1999
  - Maps Energy ⇔ Program structure (Power Profiling – Energy efficient SW design)
  - DMM gets energy for machine
  - kernel modification (system monitor) gets PIDs for processes and identifies procedures for profiling offline

# RELATED WORK – power model

- V. Tiwari, S. Malik, and A. Wolfe, “Power analysis of embedded software: A first step towards software power minimization”, International Conference on Computer-Aided Design & IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 1994
  - PIONEER WORK in Power Measurement/Modeling
  - Measure current drawn by an Intel 486DX2 Processor and DRAM
  - Generate Energy cost table for instructions
  - Identify inter-instructions effects: circuit state overhead, resource constraint effect, cache miss effects
  - there are 1 million like this: modeling SW energy, I won’t put here
- Lee, A. Ermedahl, and S. Min. “An accurate instruction-level energy consumption model for embedded risc processors” ACM SIGPLAN Conf. on Languages, Compilers, and Tools for Embedded Systems (LCTES'01), Jun 2001
  - Derives energy consumption for instructions rather than functional units for RISC ARM7TDMI processor
  - Uses their cycle-accurate power measurement scheme
  - Black box approach (similar to F. Bellosa) with linear regression

# RELATED WORK – power model

- J. Russell and M.F. Jacome, "Software Power Estimation and Optimization for High Performance, 32-bit Embedded Processors," Proc. of ICCD '98
  - Estimates SW energy for i960 family 32 bit embedded RISC processors
  - Uses digitizing oscilloscope/series Resistor  over processor power lines for measurement
  - Uses const Pest for processor power and estimates energy based on runtime ( won't work with clock gating!)
- J. Haid, G. Kafer, et al, "Run-Time Energy Estimation in System-On-a-Chip Designs", ASP-DAC 2003
  - Proposes a coprocessor for runtime energy estimation for SoC
  - Defines similar event counters in coprocessor and uses power macro-models
- M. Lajolo, A. Raghunathan, S. Dey, L. Lavagno, and A. Sangiovanni-Vincentelli. "Efficient power estimation techniques for hw/sw systems", IEEE Proc. VOLTA'99 International Workshop on Low Power Design, pages 191--199, March 1999.
  - Power estimation for HW/SW SoC designs
  - RTL HW simulator and Instruction Set simulator using instruction level power models

# RELATED WORK – power model

- M. Huang, J. Renau, and J. Torrellas. “Profile-based energy reduction in high-performance processors”, In 4th Workshop on Feedback-Directed and Dynamic Optimization, December 2001
  - Use profiling to determine when to activate/deactivate low power methods –i.e. DVS, clock gating, etc.
  - Use energy statistics (power breakdowns) from performance counters for profiling (SIM)
- I. Kadayif , T. Chinoda , M. Kandemir , N. Vijaykirsnan , M. J. Irwin , A. Sivasubramaniam, “vEC: virtual energy counters”, Proceedings of the 2001 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering, 2001
  - Uses Perfmon library for UltraSPARC to read SPARC HW perf counters related to memory
  - Converts readings to power using analytical memory energy model
  - estimates memory system energy consumption

# RELATED WORK – power model

- Luca Benini et al
  - "System-level power estimation and optimization", Proceedings 1998 international symposium on Low power electronics and design
  - "System-level power optimization: techniques and tools", Proceedings of international symposium on Low power electronics and design, 1999
    - Tutorial on power conscious system level design
    - Memory optimizations, Hardware software partitioning, instruction level power optimizations, DVS, DPM (allow components to sleep)
  - "Supporting system-level power exploration for DSP applications", Proceedings of the 10th Great Lakes Symposium on VLSI, 2000
    - Modified ARM simulator for instruction level power estimation

# RELATED WORK – thermal model

- K. Skadron, T. Abdelzaher, and M. R. Stan. “Control-theoretic techniques and thermal-RC modeling for accurate and localized dynamic thermal management”, In Proc. HPCA-8, pages 17--28, Feb. 2002.
  - Single degree component based thermal R-C model for MIPS R10000 scaled to 0.18Um
  - Only die → heatsink thermal conduction, with const. heatsink and Si properties only
  - Power/Thermal Simulation using Wattch for verification of DTM with PID controller
- Sabry, M.-N.; Bontemps, A.; Aubert, V.; Vahrmann, R, “Realistic and efficient simulation of electro-thermal effects in VLSI circuits”, Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , Volume: 5 Issue: 3 , Sep 1997
  - Transistor level with interdevice thermal resistances
- Szekely, V.; Poppe, A.; Pahi, A.; Csendes, A.; Hajas, G.; Rencz, M, “Electro-thermal and logi-thermal simulation of VLSI designs”, Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , Volume: 5 Issue: 3 , Sep 1997
  - LOGITHERM simulator module for gate level thermal simulation, by thermal characterization of logic gates

# RELATED WORK – thermal model

- COSMOS/FloWorks by NIKA
  - fluid flow and thermal analysis program
  - Heat flow computation based on mesh analysis
- A. Dhodapkar, C. H. Lim, G. Cai, and W. R. Daasch. “TEMPEST: A thermal enabled multi-model power / performance estimator”, Proceedings of Workshop on Power-Aware Computer Systems, Nov. 2000.
  - Thermally enabled architectural simulator based on SimpleScalar
  - Single R,C for the whole processor → packaging oriented
- D. Brooks and M. Martonosi. *Dynamic thermal management for high-performance microprocessors*. In Proceedings of the Seventh International Symposium on High-Performance Computer Architecture, pages 171--82, Jan. 2001.
  - Discusses Microarchitectural and scaling DTM mechanisms
  - Uses moving average of power for ~100K cycles of Wattch simulation as a proxy for temperature to detect thermal emergencies for DTM triggering

# RELATED WORK – thermal model

- Thermal Monitoring, “Intel Architecture SW developer’s Manual vol. 3”
  - Catastrophic shutdown detector
    - thermal diode resets stop clock duty cycle
  - Automatic Thermal monitor
    - Internally modulate stop clock duty cycle
  - Software controlled clock modulation
    - SW modulates stop clock duty cycle
- Kevin Skadron et al, “Temperature aware Microarchitecture”, 30th ISCA, 2003
  - HotSpot: architecture level thermal simulator built upon Wattch
  - Uses multiple degree thermal R-C model for die, packaging, heatsink and convection to ambient
  - More realistic area estimates based on Alpha 21364

# Conclusions

- Orion is a useful tool/model for rapidly exploring power-performance tradeoffs in network micro-architecure design